

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Applicants: Takashi YOKOKAWA et al.  
International Application No.: PCT/JP04/005562  
International Filing Date: April 19, 2004  
For: DECODING APPARATUS, DECODING METHOD,  
AND PROGRAM

745 Fifth Avenue  
New York, NY 10151

**EXPRESS MAIL**

Mailing Label Number: EV206809919US

Date of Deposit: January 12, 2005

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" Service under 37 CFR 1.10 on the date indicated above and is addressed to Mail Stop PCT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Adam Ahmed  
(Typed or printed name of person mailing paper or fee)

A. Ahmed  
(Signature of person mailing paper or fee)

**CLAIM OF PRIORITY UNDER 37 C.F.R. § 1.78(a)(2)**

Mail Stop PCT  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

Pursuant to 35 U.S.C. 119, this application is entitled to a claim of priority to Japan  
Application Nos. 2003-133941 and 2003-294383 filed 13 May and 18 August 2003.

Respectfully submitted,

FROMMER LAWRENCE & HAUG LLP  
Attorneys for Applicants

By: William S. Frommer  
William S. Frommer  
Reg. No. 25,506  
Tel. (212) 588-0800

**BEST AVAILABLE COPY**

00243120

10/52105A

Rec'd PCT/PTO 12 JAN 2005

#2

日本国特許庁  
JAPAN PATENT OFFICE

19.4.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。  
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 8月18日  
Date of Application:

出願番号 特願2003-294383  
Application Number:  
[ST. 10/C]: [JP 2003-294383]

出願人 ソニー株式会社  
Applicant(s):

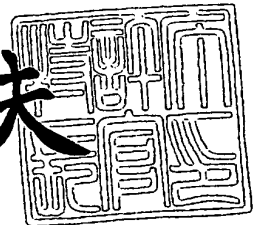
REC'D 13 MAY 2004  
WIPO PCT

PRIORITY DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

2004年 2月19日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



出証番号 出証特2004-3011373

【書類名】 特許願  
【整理番号】 0390471103  
【提出日】 平成15年 8月18日  
【あて先】 特許庁長官殿  
【国際特許分類】 H03M 13/03  
【発明者】  
    【住所又は居所】 東京都品川区北品川 6丁目 7番 35号 ソニー株式会社内  
    【氏名】 横川 峰志  
【発明者】  
    【住所又は居所】 東京都品川区北品川 6丁目 7番 35号 ソニー株式会社内  
    【氏名】 宮内 俊之  
【発明者】  
    【住所又は居所】 東京都品川区北品川 6丁目 7番 35号 ソニー株式会社内  
    【氏名】 飯田 康博  
【特許出願人】  
    【識別番号】 000002185  
    【氏名又は名称】 ソニー株式会社  
【代理人】  
    【識別番号】 100082131  
    【弁理士】  
    【氏名又は名称】 稲本 義雄  
    【電話番号】 03-3369-6479  
【先の出願に基づく優先権主張】  
    【出願番号】 特願2003-133941  
    【出願日】 平成15年 5月13日  
【手数料の表示】  
    【予納台帳番号】 032089  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 9708842

**【書類名】 特許請求の範囲****【請求項 1】**

LDPC (Low Density Parity Check) 符号の復号装置であって、  
P×Pの単位行列、その単位行列のコンポーネントである1のうちの1個以上が0になった行列である準単位行列、前記単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、前記単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、またはP×Pの0行列を構成行列として、前記LDPC符号の検査行列が、複数の前記構成行列の組合せで表される場合において、

前記LDPC符号の復号のためのP個のチェックノードの演算を同時に行う第1の演算手段と、

前記LDPC符号の復号のためのP個のバリエブルノードの演算を同時に行う第2の演算手段と

を備えることを特徴とする復号装置。

**【請求項 2】**

請求項1に記載の復号装置であって、

前記第1の演算手段は、チェックノードの演算を行うP個のチェックノード計算器を有し、

前記第2の演算手段は、バリエブルノードの演算を行うP個のバリエブルノード計算器を有する

ことを特徴とする復号装置。

**【請求項 3】**

請求項1に記載の復号装置であって、

前記P個のチェックノードの演算、または前記P個のバリエブルノードの演算の結果得られるP個の枝に対応するメッセージデータを同時に読み書きするメッセージ記憶手段をさらに備える

ことを特徴とする復号装置。

**【請求項 4】**

請求項3に記載の復号装置であって、

前記メッセージ記憶手段は、チェックノードの演算時に読み出される枝に対応するメッセージデータを、検査行列の1を行方向に詰めるように格納する

ことを特徴とする復号装置。

**【請求項 5】**

請求項3に記載の復号装置であって、

前記メッセージ記憶手段は、バリエブルノード演算時に読み出される枝に対応するメッセージデータを、検査行列の1を列方向に詰めるように格納する

ことを特徴とする復号装置。

**【請求項 6】**

請求項3に記載の復号装置であって、

前記メッセージ記憶手段は、前記検査行列を表す構成行列のうちの、重みが2以上の構成行列について、その構成行列を、重みが1の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列に属するP個の枝に対応するメッセージを、同一のアドレスに格納する

ことを特徴とする復号装置。

**【請求項 7】**

請求項3に記載の復号装置であって、

前記メッセージ記憶手段は、行数/P個のFIFOと、列数/P個のFIFOとで構成され、

前記行数/P個のFIFOと列数/P個のFIFOは、それぞれ、前記検査行列の行と列の重みに対応するワード数を有する

ことを特徴とする復号装置。

**【請求項 8】**

請求項 3 に記載の復号装置であって、  
前記メッセージ記憶手段は、RAM(Random Access Memory)で構成され、  
前記RAMは、前記メッセージデータを、読み出される順番に詰めて格納し、格納位置順  
に読み出す  
ことを特徴とする復号装置。

【請求項 9】

請求項 1 に記載の復号装置であって、  
LDPC符号の受信情報を格納するとともに、P個の前記受信情報を同時に読み出す受信情  
報記憶手段をさらに備える  
ことを特徴とする復号装置。

【請求項 10】

請求項 9 に記載の復号装置であって、  
前記受信情報記憶手段は、前記受信情報を、前記バリエブルノードの演算に必要となる  
順番に読み出すことができるように格納する  
ことを特徴とする復号装置。

【請求項 11】

請求項 1 に記載の復号装置であって、  
前記P個のチェックノードの演算、または前記P個のバリエブルノードの演算の結果得ら  
れるメッセージを並べ替える並べ替え手段をさらに備える  
ことを特徴とする復号装置。

【請求項 12】

請求項 11 に記載の復号装置であって、  
前記並べ替え手段は、バレルシフタで構成される  
ことを特徴とする復号装置。

【請求項 13】

請求項 1 に記載の復号装置であって、  
前記第 1 の演算手段と前記第 2 の演算手段は、P個の枝に対応するメッセージを求める  
ことを特徴とする復号装置。

【請求項 14】

請求項 1 に記載の復号装置であって、  
前記第 1 の演算手段は、前記P個のチェックノードの演算と前記P個のバリエブルノード  
の演算の一部とを行い、  
前記第 2 の演算手段は、前記P個のバリエブルノードの演算の他の一部を行う  
ことを特徴とする復号装置。

【請求項 15】

請求項 14 に記載の復号装置であって、  
前記第 1 の演算手段は、前記P個のチェックノードの演算と前記P個のバリエブルノード  
の演算の一部を行うP個の計算器を有し、  
前記第 2 の演算手段は、前記P個のバリエブルノードの演算の他の一部を行うP個の計算  
器を有する  
ことを特徴とする復号装置。

【請求項 16】

請求項 14 に記載の復号装置であって、  
前記第 1 の演算手段が前記P個のチェックノードの演算と前記P個のバリエブルノードの  
演算の一部を行うことにより得られるP個の枝に対応する第 1 の復号途中結果を同時に読  
み書きする第 1 の復号途中結果記憶手段をさらに備える  
ことを特徴とする復号装置。

【請求項 17】

請求項 16 に記載の復号装置であって、  
前記第 1 の復号途中記憶手段は、前記P個のバリエブルノードの演算の他の一部を行う

時に読み出される枝に対応する前記第1の復号途中結果を、検査行列の1を行方向に詰め  
るように格納する

ことを特徴とする復号装置。

【請求項18】

請求項16に記載の復号装置であって、

前記第1の復号途中結果記憶手段は、2個のシングルポートRAM(Random Access Memory)  
)である

ことを特徴とする復号装置。

【請求項19】

請求項18に記載の復号装置であって、

前記2個のシングルポートRAMは、前記第1の復号途中結果を前記検査行列のP行の枝に  
対応する前記第1の復号途中結果ずつ交互に格納する

ことを特徴とする復号装置。

【請求項20】

請求項18に記載の復号装置であって、

前記2個のシングルポートRAM(Random Access Memory)は、それぞれ同一のアドレスに  
格納している前記第1の復号途中結果を読み出す

ことを特徴とする復号装置。

【請求項21】

請求項16記載の復号装置であって、

前記第1の復号途中結果記憶手段は、前記検査行列を表す構成行列のうちの、重みが2  
以上の構成行列について、その構成行列を、重みが1の単位行列、準単位行列、またはシ  
フト行列の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフ  
ト行列に属するP個の枝に対応する前記第1の復号途中結果を、同一のアドレスに格納す  
る

ことを特徴とする復号装置。

【請求項22】

請求項14に記載の復号装置であって、

前記第2の演算手段が前記P個のバリエブルノードの演算の他の一部を行うことにより  
得られるP個の枝に対応する前記第2の復号途中結果を同時に読み書きする第2の復号途  
中結果記憶手段をさらに備える

ことを特徴とする復号装置。

【請求項23】

請求項14に記載の復号装置であって、

LDPC符号の受信情報を格納するとともに、P個の前記受信情報を同時に読み出す受信情  
報記憶手段をさらに備える

ことを特徴とする復号装置。

【請求項24】

請求項23に記載の復号装置であって、

前記受信情報記憶手段は、前記受信情報を、前記P個のバリエブルノードの演算の他の  
一部の演算に必要な順番に読み出すことができるように格納する

ことを特徴とする復号装置。

【請求項25】

請求項14に記載の復号装置であって、

前記第1の演算手段が前記P個のチェックノードの演算と前記P個のバリエブルノードの  
演算の一部を行うことにより得られる第1の復号途中結果、または前記第2の演算手段が  
前記P個のバリエブルノードの演算の他の一部を行うことにより得られる第2の復号途中  
結果を並べ替える並べ替え手段をさらに備える

ことを特徴とする復号装置。

【請求項26】

請求項 25 に記載の復号装置であって、  
前記並べ替え手段は、バレルシフタで構成される  
ことを特徴とする復号装置。

【請求項 27】

請求項 1 に記載の復号装置であって、  
前記第 1 の演算手段は、前記 P 個のチェックノードの演算の一部を行い、  
前記第 2 の演算手段は、前記 P 個のチェックノードの演算の他の一部と、前記 P 個のバリ  
アブルノードの演算とを行う  
ことを特徴とする復号装置。

【請求項 28】

請求項 27 に記載の復号装置であって、  
前記第 1 の演算手段は、前記 P 個のチェックノードの演算の一部を行う P 個の計算器を有  
し、  
前記第 2 の演算手段は、前記 P 個のチェックノードの演算の他の一部と、前記 P 個のバリ  
アブルノードの演算を行う P 個の計算器を有する  
ことを特徴とする復号装置。

【請求項 29】

請求項 27 に記載の復号装置であって、  
前記第 1 の演算手段が前記 P 個のチェックノードの演算の一部を行うことにより得られ  
る P 個の枝に対応する第 1 の復号途中結果を同時に読み書きする第 1 の復号途中結果記憶  
手段をさらに備える  
ことを特徴とする復号装置。

【請求項 30】

請求項 27 に記載の復号装置であって、  
前記第 2 の演算手段が前記 P 個のチェックノードの演算の他の一部と、前記 P 個のバリ  
アブルノードの演算を行うことにより得られる P 個の枝に対応する第 2 の復号途中結果を同  
時に読み書きする第 2 の復号途中結果記憶手段をさらに備える  
ことを特徴とする復号装置。

【請求項 31】

請求項 30 に記載の復号装置であって、  
前記第 2 の復号途中結果記憶手段は、前記 P 個のチェックノードの演算の他の一部と、  
前記 P 個のバリアブルノードの演算を行う時に読み出される枝に対応する前記第 2 の復号  
途中結果を、検査行列の 1 を列方向に詰めるように格納する  
ことを特徴とする復号装置。

【請求項 32】

請求項 30 に記載の復号装置であって、  
前記第 2 の復号途中結果記憶手段は、2 個のシングルポート RAM(Random Access Memory  
)である  
ことを特徴とする復号装置。

【請求項 33】

請求項 32 に記載の復号装置であって、  
前記 2 個のシングルポート RAM は、前記第 2 の復号途中結果を前記検査行列の P 列の枝  
に対応する前記第 2 の復号途中結果ずつ交互に格納する  
ことを特徴とする復号装置。

【請求項 34】

請求項 32 に記載の復号装置であって、  
前記 2 個のシングルポート RAM(Random Access Memory) は、それぞれ同一のアドレスに  
格納している前記第 2 の復号途中結果を読み出す  
ことを特徴とする復号装置。

【請求項 35】

請求項 30 に記載の復号装置であって、  
前記第 2 の復号途中結果記憶手段は、前記検査行列を表す構成行列のうちの、重みが 2 以上の構成行列について、その構成行列を、重みが 1 の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが 1 の単位行列、準単位行列、またはシフト行列に属する P 個の枝に対応する前記第 2 の復号途中結果を、同一のアドレスに格納する

ことを特徴とする復号装置。

【請求項 36】

請求項 27 に記載の復号装置であって、  
LDPC 符号の受信情報を格納するとともに、P 個の前記受信情報を同時に読み出す受信情報記憶手段をさらに備える

ことを特徴とする復号装置。

【請求項 37】

請求項 36 に記載の復号装置であって、  
前記受信情報記憶手段は、前記受信情報を、前記 P 個のチェックノードの演算の他の一部と、前記 P 個のバリエブルノードの演算に必要となる順番に読み出すことができるように格納する

ことを特徴とする復号装置。

【請求項 38】

請求項 27 に記載の復号装置であって、  
前記第 1 の演算手段が前記 P 個のチェックノードの演算の一部を行うことにより得られる第 1 の復号途中結果、または前記第 2 の演算が前記 P 個のチェックノードの演算の他の一部と、前記 P 個のバリエブルノードの演算を行うことにより得られる第 2 の復号途中結果を並べ替える並べ替え手段をさらに備える

ことを特徴とする復号装置。

【請求項 39】

請求項 38 に記載の復号装置であって、  
前記並べ替え手段は、バレルシフタで構成される

ことを特徴とする復号装置。

【請求項 40】

LDPC (Low Density Parity Check) 符号の復号装置の復号方法であって、  
P×P の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上が 0 になった行列である準単位行列、前記単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、前記単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または P×P の 0 行列を構成行列として、前記 LDPC 符号の検査行列が、複数の前記構成行列の組合せで表される場合において、

前記 LDPC 符号の復号のための P 個のチェックノードの演算を同時に行う第 1 の演算ステップと、

前記 LDPC 符号の復号のための P 個のバリエブルノードの演算を同時に行う第 2 の演算ステップと

を含むことを特徴とする復号方法。

【請求項 41】

LDPC (Low Density Parity Check) 符号の復号をコンピュータに行わせるプログラムであって、

前記 LDPC 符号の復号のための P 個のチェックノードの演算を同時に行う第 1 の演算ステップと、

前記 LDPC 符号の復号のための P 個のバリエブルノードの演算を同時に行う第 2 の演算ステップと

を含むことを特徴とするプログラム。



【書類名】明細書

【発明の名称】復号装置および復号方法、並びにプログラム

【技術分野】

【0001】

本発明は、復号装置および復号方法、並びにプログラムに関し、特に、低密度パリティ検査符号(LDPC符号)による符号化が施された符号の復号を行う復号装置および復号方法、並びにプログラムに関する。

【背景技術】

【0002】

近年、例えば、移動体通信や深宇宙通信といった通信分野、及び地上波又は衛星デジタル放送といった放送分野の研究が著しく進められているが、それに伴い、誤り訂正符号化及び復号の効率化を目的として符号理論に関する研究も盛んに行われている。

【0003】

符号性能の理論的限界としては、いわゆるシャノン(C. E. Shannon)の通信路符号化定理によって与えられるシャノン限界が知られている。符号理論に関する研究は、このシャノン限界に近い性能を示す符号を開発することを目的として行われている。近年では、シャノン限界に近い性能を示す符号化方法として、例えば、並列連接畳み込み符号(PCCC(Parallel Concatenated Convolutional Codes))や、縦列連接畳み込み符号(SCCC(Serially Concatenated Convolutional Codes))といった、いわゆるターボ符号化(Turbo coding)と呼ばれる手法が開発されている。また、これらのターボ符号が開発される一方で、古くから知られる符号化方法である低密度パリティ検査符号(Low Density Parity Check codes)(以下、LDPC符号という)が脚光を浴びつつある。

【0004】

LDPC符号は、R. G. Gallagerによる「R. G. Gallager, "Low Density Parity Check Codes", Cambridge, Massachusetts: M. I. T. Press, 1963」において最初に提案されたものであり、その後、「D. J. C. MacKay, "Good error correcting codes based on very sparse matrices", Submitted to IEEE Trans. Inf. Theory, IT-45, pp. 399-431, 1999」や、「M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi and D. A. Spielman, "Analysis of low density codes and improved designs using irregular graphs", in Proceedings of ACM Symposium on Theory of Computing, pp. 249-258, 1998」等において再注目されるに至ったものである。

【0005】

LDPC符号は、近年の研究により、ターボ符号等と同様に、符号長を長くしていくにしたがって、シャノン限界に近い性能が得られることがわかりつつある。また、LDPC符号は、最小距離が符号長に比例するという性質があることから、その特徴として、ブロック誤り確率特性がよく、さらに、ターボ符号等の復号特性において観測される、いわゆるエラーフロア現象が殆ど生じないことも利点として挙げられる。

【0006】

以下、このようなLDPC符号について具体的に説明する。なお、LDPC符号は、線形符号であり、必ずしも2元である必要はないが、ここでは、2元であるものとして説明する。

【0007】

LDPC符号は、そのLDPC符号を定義する検査行列(parity check matrix)が疎なものであることを最大の特徴とするものである。ここで、疎な行列とは、行列のコンポーネントの"1"の個数が非常に少なく構成されるものであり、疎な検査行列をHで表すものとする、そのような検査行列としては、例えば、図1に示すように、各列のハミング重み("1"の数)(weight)が"3"であり、且つ、各行のハミング重みが"6"であるもの等がある。

【0008】

このように、各行及び各列のハミング重みが一定である検査行列Hによって定義されるLDPC符号は、レギュラーLDPC符号と称される。一方、各行及び各列のハミング重みが一定

でない検査行列Hによって定義されるLDPC符号は、イレギュラーLDPC符号と称される。

#### 【0009】

このようなLDPC符号による符号化は、検査行列Hに基づいて生成行列Gを生成し、この生成行列Gを2元の情報メッセージに対して乗算することによって符号語を生成することで実現される。具体的には、LDPC符号による符号化を行う符号化装置は、まず、検査行列Hの転置行列 $H^T$ との間に、式 $GH^T=0$ が成立する生成行列Gを算出する。ここで、生成行列Gが、 $k \times n$ 行列である場合には、符号化装置は、生成行列Gに対してkビットからなる情報メッセージ（ベクトルu）を乗算し、nビットからなる符号語c（ $=uG$ ）を生成する。この符号化装置によって生成された符号語は、値が“0”の符号ビットが“+1”に、値が“1”の符号ビットが“-1”にといったようにマッピングされて送信され、所定の通信路を介して受信側において受信されることになる。

#### 【0010】

一方、LDPC符号の復号は、Gallagerが確率復号(Probabilistic Decoding)と称して提案したアルゴリズムであって、バリアブルノード(variable node (メッセージノード(message node)とも呼ばれる。))と、チェックノード(check node)とからなる、いわゆるタナグラフ(Tanner graph)上での確率伝播(belief propagation)によるメッセージ・パッシング・アルゴリズムによって行うことが可能である。ここで、以下、適宜、バリアブルノードとチェックノードを、単に、ノードともいう。

#### 【0011】

しかしながら、確率復号においては、各ノード間で受け渡されるメッセージが実数値であることから、解析的に解くためには、連続した値をとるメッセージの確率分布そのものを追跡する必要があり、非常に困難を伴う解析を必要とすることになる。そこで、Gallagerは、LDPC符号の復号アルゴリズムとして、アルゴリズムA又はアルゴリズムBを提案している。

#### 【0012】

LDPC符号の復号は、一般的には、図2に示すような手順にしたがって行われる。なお、ここでは、受信値を $U_0(u_{0i})$ とし、チェックノードから出力されるメッセージを $u_j$ とし、バリアブルノードから出力されるメッセージを $v_i$ とする。また、ここでは、メッセージとは、値の“0”らしさを、いわゆる対数尤度比(log likelihood ratio)で表現した実数値である。

#### 【0013】

まず、LDPC符号の復号においては、図2に示すように、ステップS11において、受信値 $U_0(u_{0i})$ が受信され、メッセージ $u_j$ が“0”に初期化されるとともに、繰り返し処理のカウントとしての整数をとる変数kが“0”に初期化され、ステップS12に進む。ステップS12において、受信値 $U_0(u_{0i})$ に基づいて、式(1)に示す演算を行うことによってメッセージ $v_i$ が求められ、さらに、このメッセージ $v_i$ に基づいて、式(2)に示す演算を行うことによってメッセージ $u_j$ が求められる。

#### 【0014】

##### 【数1】

$$v_i = u_{0i} + \sum_{j=1}^{d_v-1} u_j \quad \dots (1)$$

#### 【0015】

##### 【数2】

$$\tanh\left(\frac{u_j}{2}\right) = \prod_{i=1}^{d_c-1} \tanh\left(\frac{v_i}{2}\right) \quad \dots (2)$$

#### 【0016】

ここで、式(1)と式(2)における $d_v$ と $d_c$ は、それぞれ、検査行列Hの縦方向(行方向)と横方向(列方向)の"1"の個数を示す任意に選択可能とされるパラメータであり、例えば、(3,6)符号の場合には、 $d_v=3$ ,  $d_c=6$ となる。

【0017】

なお、式(1)または(2)の演算においては、それぞれ、メッセージを出力しようとする枝(edge)から入力されたメッセージを、和または積演算のパラメータとしては用いないことから、和または積演算の範囲が、1乃至 $d_v-1$ または1乃至 $d_c-1$ となっている。また、式(2)に示す演算は、実際には、2入力 $v_1$ ,  $v_2$ に対する1出力で定義される式(3)に示す関数 $R(v_1, v_2)$ のテーブルを予め作成しておき、これを式(4)に示すように連続的(再帰的)に用いることによって行われる。

【0018】

【数3】

$$x = 2 \tanh^{-1} \{ \tanh(v_1/2) \tanh(v_2/2) \} = R(v_1, v_2) \quad \dots (3)$$

【0019】

【数4】

$$u_j = R(v_1, R(v_2, R(v_3, \dots R(v_{d_c-2}, v_{d_c-1}))) \quad \dots (4)$$

【0020】

ステップS12では、さらに、変数 $k$ が"1"だけインクリメントされ、ステップS13に進む。ステップS13では、変数 $k$ が所定の繰り返し復号回数 $N$ 以上であるか否かが判定される。ステップS13において、変数 $k$ が $N$ 以上ではないと判定された場合、ステップS12に戻り、以下、同様の処理が繰り返される。

【0021】

また、ステップS13において、変数 $k$ が $N$ 以上であると判定された場合、ステップS14に進み、式(5)に示す演算を行うことによって最終的に出力する復号結果としてのメッセージ $v$ が求められて出力され、LDPC符号の復号処理が終了する。

【0022】

【数5】

$$v = u_{0i} + \sum_{j=1}^{d_v} u_j \quad \dots (5)$$

【0023】

ここで、式(5)の演算は、式(1)の演算とは異なり、バリエブルノードに接続している全ての枝からの入力メッセージを用いて行われる。

【0024】

このようなLDPC符号の復号は、例えば(3,6)符号の場合には、図3に示すように、各ノード間でメッセージの授受が行われる。なお、図3における"="で示すノード(バリエブルノード)では、式(1)に示した演算が行われ、"+"で示すノード(チェックノード)では、式(2)に示した演算が行われる。特に、アルゴリズムAにおいては、メッセージを2元化し、"+"で示すノードにて、 $d_c-1$ 個の入力メッセージの排他的論理和演算を行い、"="で示すノードにて、受信値 $R$ に対して、 $d_v-1$ 個の入力メッセージが全て異なるビット値であった場合には、符号を反転して出力する。

【0025】

また、一方で、近年、LDPC符号の復号の実装法に関する研究も行われている。実装方法について述べる前に、まず、LDPC符号の復号を模式化して説明する。

【0026】

図4は、(3,6)LDPC符号(符号化率1/2、符号長12)の検査行列(parity check matrix)の例である。LDPC符号の検査行列は、図5のように、タナーグラフを用いて書き表すことができる。ここで、図5において、“+”で表わされるのが、チェックノードであり、“=”で表わされるのが、バリエブルノードである。チェックノードとバリエブルノードは、それぞれ、検査行列の行と列に対応する。チェックノードとバリエブルノードとの間の結線は、枝(edge)であり、検査行列の“1”に相当する。即ち、検査行列の第j行第i列のコンポネントが1である場合には、図5において、上からi番目のバリエブルノード(“=”のノード)と、上からj番目のチェックノード(“+”のノード)とが、枝により接続される。枝は、バリエブルノードに対応する符号ビットが、チェックノードに対応する拘束条件を持つことを表わす。なお、図5は、図4の検査行列のタナーグラフとなっている。

## 【0027】

LDPC符号の復号方法であるサンプロダクトアルゴリズム(Sum Product Algorithm)は、バリエブルノードの演算とチェックノードの演算とを繰り返す行う。

## 【0028】

バリエブルノードでは、図6のように、式(1)の演算を行う。すなわち、図6において、計算しようとしている枝に対応するメッセージ $v_i$ は、バリエブルノードに繋がっている残りの枝からのメッセージ $u_1$ および $u_2$ と、受信情報 $u_{0i}$ を用いて計算される。他の枝に対応するメッセージも同様に計算される。

## 【0029】

チェックノードの演算について説明する前に、式(2)を、式 $a \times b = \exp\{\ln(|a|) + \ln(|b|)\} \times \text{sign}(a) \times \text{sign}(b)$ の関係を用いて、式(6)のように書き直す。但し、 $\text{sign}(x)$ は、 $x \geq 0$ のとき1であり、 $x < 0$ のとき-1である。

## 【0030】

## 【数6】

$$\begin{aligned} u_j &= 2 \tanh^{-1} \left( \prod_{i=1}^{d_c-1} \tanh \left( \frac{v_i}{2} \right) \right) \\ &= 2 \tanh^{-1} \left[ \exp \left\{ \sum_{i=1}^{d_c-1} \ln \left( \left| \tanh \left( \frac{v_i}{2} \right) \right| \right) \right\} \times \prod_{i=1}^{d_c-1} \text{sign} \left( \tanh \left( \frac{v_i}{2} \right) \right) \right] \\ &= 2 \tanh^{-1} \left[ \exp \left\{ - \left( \sum_{i=1}^{d_c-1} - \ln \left( \tanh \left( \frac{|v_i|}{2} \right) \right) \right) \right\} \times \prod_{i=1}^{d_c-1} \text{sign}(v_i) \right] \end{aligned} \quad \dots (6)$$

## 【0031】

更に、 $x \geq 0$ において、 $\phi(x) = \ln(\tanh(x/2))$ と定義すると、 $\phi^{-1}(x) = 2 \tanh^{-1}(e^{-x})$ であるから、式(6)は、式(7)のように書くことができる。

## 【0032】

## 【数7】

$$u_j = \phi^{-1} \left( \sum_{i=1}^{d_c-1} \phi(|v_i|) \right) \times \prod_{i=1}^{d_c-1} \text{sign}(v_i)$$

... (7)

## 【0033】

チェックノードでは、図7のように、式(7)の演算を行う。すなわち、図7において、計算しようとしている枝に対応するメッセージ $u_j$ は、チェックノードに繋がっている残りの枝からのメッセージ $v_1, v_2, v_3, v_4, v_5$ を用いて計算される。他の枝に対応するメッセー

ジも同様に計算される。

#### 【0034】

なお、関数  $\phi(x)$  は、 $\phi(x) = \ln((e^x + 1)/(e^x - 1))$  とも表すことができ、 $x > 0$  において、 $\phi(x) = \phi^{-1}(x)$  である。関数  $\phi(x)$  および  $\phi^{-1}(x)$  をハードウェアに実装する際には、LUT (Look Up Table) を用いて実装される場合があるが、両者共に同一の LUT となる。

#### 【0035】

サンプラダクトアルゴリズムをハードウェアに実装する場合、式 (1) で表わされるバリエブルノード演算および式 (7) で表わされるチェックノード演算とを、適度な回路規模と動作周波数で繰り返し行うことが必要である。

#### 【0036】

復号装置の実装の例として、まず、単純に各ノードの演算を一つずつ順次行うことによって復号を行う場合 (full serial decoding) の実装法について説明する。

#### 【0037】

なお、ここでは、例えば、図 8 の、30 (行)  $\times$  90 (列) の検査行列で表現される符号 (符号化率 2/3、符号長 90) を復号することとする。図 8 の検査行列の 1 の数は 269 であり、従って、そのタナグラフでは、枝の数は 269 個となる。ここで、図 8 の検査行列では、0 を、“.” で表現している。

#### 【0038】

図 9 は、LDPC 符号の 1 回復号を行う復号装置の構成例を示している。

#### 【0039】

図 9 の復号装置では、その動作する 1 クロック (clock) ごとに、1 つの枝に対応するメッセージが計算される。

#### 【0040】

即ち、図 9 の復号装置は、2 つの枝用メモリ 100 および 102、1 つのチェックノード計算器 101、1 つのバリエブルノード計算器 103、1 つの受信用メモリ 104、1 つの制御部 105 からなる。

#### 【0041】

図 9 の復号装置では、枝用メモリ 100 または 102 からメッセージデータが一つずつ読み出され、そのメッセージデータを用いて、所望の枝に対応するメッセージデータが計算される。そして、その計算によって求められたメッセージデータが一つずつ後段の枝用メモリ 102 または 100 に格納されていく。繰り返し復号を行う際には、この 1 回復号を行う図 9 の復号装置を複数個縦列に接続するか、もしくは図 9 の復号装置を繰り返し用いることによって、繰り返し復号を実現する。なお、ここでは、例えば、図 9 の復号装置が複数個接続されているものとする。

#### 【0042】

枝用メモリ 100 は、前段の復号装置 (図示せず) のバリエブルノード計算器 103 から供給されるメッセージ D100 を、後段のチェックノード計算器 101 が読み出す順番に格納していく。そして、枝用メモリ 100 は、チェックノード計算のフェーズでは、メッセージ D100 を、格納してある順番通りに、メッセージ D101 として、チェックノード計算器 101 に供給する。

#### 【0043】

チェックノード計算器 101 は、制御部 105 から供給される制御信号 D106 に基づき、枝用メモリ 100 から供給されるメッセージ D101 を用いて、式 (7) に従って演算を行い、その演算によって求められたメッセージ D102 を、後段の枝用メモリ 102 に供給する。

#### 【0044】

枝用メモリ 102 は、前段のチェックノード計算器 101 から供給されるメッセージ D102 を、後段のバリエブルノード計算器 103 が読み出す順番に格納していく。そして、枝用メモリ 102 は、バリエブルノード計算のフェーズでは、メッセージ D102 を、格納してある順番通りに、メッセージ D103 として、バリエブルノード計算器 103 に供給する。

#### 【0045】

さらに、バリアブルノード計算器 103 には、制御部 105 から制御信号 D107 が供給されるとともに、受信用メモリ 104 から受信データ D104 が供給される。バリアブルノード計算器 103 は、制御信号 D107 に基づき、枝用メモリ 100 から供給されるメッセージ D103 と受信用メモリ 100 から供給される受信データ D104 を用い、式 (1) に従って演算を行い、その演算の結果得られるメッセージ D105 を、図示せぬ後段の復号装置の枝用メモリ 100 に供給する。

#### 【0046】

受信用メモリ 104 には、LDPC 符号化された受信データ (LDPC 符号) が格納される。制御部 105 は、バリアブルノード演算を制御する制御信号 D106 と、チェックノード演算を制御する制御信号 D107 を、それぞれチェックノード計算器 101 とバリアブルノード計算器 103 に供給する。制御部 105 は、枝用メモリ 100 に全ての枝のメッセージが格納されたとき、チェックノード計算器 101 に制御信号 D106 を供給し、枝用メモリ 102 に全ての枝のメッセージが格納されたとき、バリアブルノード計算器 103 に制御信号 D107 を供給する。

#### 【0047】

図 10 は、チェックノード演算を 1 つずつ行う図 9 のチェックノード計算器 101 の構成例を示している。

#### 【0048】

なお、図 10 では、各メッセージが符号ビットを合わせて合計 6 ビット (bit) に量子化されているものとして、チェックノード計算器 101 を表している。また、図 10 では、図 8 の検査行列で表わされる LDPC 符号のチェックノード演算が行われる。さらに、図 10 のチェックノード演算器 101 には、クロック  $ck$  が供給され、このクロック  $ck$  は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック  $ck$  に同期して処理を行う。

#### 【0049】

図 10 のチェックノード計算器 101 は、制御部 105 から供給される、例えば、1 ビットの制御信号 D106 に基づき、枝用メモリ 100 から 1 つずつ読み込まれるメッセージ D101 を用いて、式 (7) にしたがって演算を行う。

#### 【0050】

即ち、チェックノード計算器 101 では、検査行列の各列に対応するバリアブルノードからの 6 ビットのメッセージ D101 (メッセージ  $v_i$ ) が 1 つずつ読み込まれ、その下位ビットである絶対値 D122 ( $|v_i|$ ) が LUT 121 に、その最上位ビットである符号ビット D121 が EXOR 回路 129 と FIFO (First In First Out) メモリ 133 にそれぞれ供給される。また、チェックノード計算器 101 には、制御部 105 から制御信号 D106 が供給され、その制御信号 D106 は、セレクト 124 とセレクト 131 に供給される。

#### 【0051】

LUT 121 は、絶対値 D122 ( $|v_i|$ ) に対して、式 (7) における  $\phi(|v_i|)$  の演算を行った 5 ビットの演算結果 D123 ( $\phi(|v_i|)$ ) を読み出し、加算器 122 と FIFO メモリ 127 に供給する。

#### 【0052】

加算器 122 は、演算結果 D123 ( $\phi(|v_i|)$ ) とレジスタ 123 に格納されている 9 ビットの値 D124 とを加算することにより、演算結果 D123 を積算し、その結果得られる 9 ビットの積算値をレジスタ 123 に再格納する。なお、検査行列の 1 行に亘る全ての枝からのメッセージ D101 の絶対値 D122 ( $|v_i|$ ) に対する演算結果が積算された場合、レジスタ 123 はリセットされる。

#### 【0053】

検査行列の 1 行に亘るメッセージ D101 が 1 つずつ読み込まれ、レジスタ 123 に 1 行分の演算結果 D123 が積算された積算値が格納された場合、制御部 105 から供給される制御信号 D106 は、0 から 1 に変化する。例えば、行の重み (row weight) が「9」である場合、制御信号 D106 は、1 から 8 クロック目までは、「0」となり、9 クロック目では「1」と

なる。

#### 【0054】

制御信号D106が「1」の場合、セクタ124は、レジスタ123に格納されている値、即ち、検査行列の1行に亘る全ての枝からのメッセージD101（メッセージ $v_i$ ）から求められた $\phi(|v_i|)$ が積算された9ビットの値D124（ $i=1$ から $i=d_c$ までの $\Sigma \phi(|v_i|)$ ）を選択し、値D125として、レジスタ125に出力して格納させる。レジスタ125は、格納している値D125を、9ビットの値D126として、セクタ124と加算器126に供給する。制御信号D106が「0」の場合、セクタ124は、レジスタ125から供給された値D126を選択し、レジスタ125に出力して再格納させる。即ち、検査行列の1行に亘る全ての枝からのメッセージD101（メッセージ $v_i$ ）から求められた $\phi(|v_i|)$ が積算されるまで、レジスタ125は、前回積算された $\phi(|v_i|)$ を、セクタ124と加算器126に供給する。

#### 【0055】

一方、FIFOメモリ127は、レジスタ125から新たな値D126（ $i=1$ から $i=d_c$ までの $\Sigma \phi(|v_i|)$ ）が出力されるまでの間、LUT121が出力した演算結果D123（ $\phi(|v_i|)$ ）を遅延し、5ビットの値D127として減算器126に供給する。減算器126は、レジスタ125から供給された値D126から、FIFOメモリ127から供給された値D127を減算し、その減算結果を、5ビットの減算値D128としてLUT128に供給する。即ち、減算器126は、検査行列の1行に亘る全ての枝からのメッセージD101（メッセージ $v_i$ ）から求められた $\phi(|v_i|)$ の積算値から、求めたい枝からのメッセージD101（メッセージ $v_i$ ）から求められた $\phi(|v_i|)$ を減算して、その減算値（ $i=1$ から $i=d_c-1$ までの $\Sigma \phi(|v_i|)$ ）を減算値D128としてLUT128に供給する。

#### 【0056】

LUT128は、減算値D128（ $i=1$ から $i=d_c-1$ までの $\Sigma \phi(|v_i|)$ ）に対して、式（7）における $\phi^{-1}(\Sigma \phi(|v_i|))$ の演算を行った5ビットの演算結果D129（ $\phi^{-1}(\Sigma \phi(|v_i|))$ ）を出力する。

#### 【0057】

以上の処理と並行して、EXOR回路129は、レジスタ130に格納されている1ビットの値D131と符号ビットD121との排他的論理和を演算することにより、符号ビットどうしの乗算を行い、1ビットの乗算結果D130をレジスタ130に再格納する。なお、検査行列の1行に亘る全ての枝からのメッセージD101の符号ビットD121が乗算された場合、レジスタ130はリセットされる。

#### 【0058】

検査行列の1行に亘る全ての枝からのメッセージD101の符号ビットD121が乗算された乗算結果D130（ $i=1$ から $d_c$ までの $\Pi \text{sign}(v_i)$ ）がレジスタ130に格納された場合、制御部105から供給される制御信号D106は、「0」から「1」に変化する。

#### 【0059】

制御信号D106が「1」の場合、セクタ131は、レジスタ130に格納されている値、即ち、検査行列の1行に亘る全ての枝からのメッセージD101の符号ビットD121が乗算された値D131（ $i=1$ から $i=d_c$ までの $\Pi \text{sign}(v_i)$ ）を選択し、1ビットの値D132としてレジスタ132に出力して格納させる。レジスタ132は、格納している値D132を、1ビットの値D133としてセクタ131とEXOR回路134に供給する。制御信号D106が「0」の場合、セクタ131は、レジスタ132から供給された値D133を選択し、レジスタ132に出力して再格納させる。即ち、検査行列の1行に亘る全ての枝からのメッセージD101（メッセージ $v_i$ ）の符号ビットD121が乗算されるまで、レジスタ132は、前回格納した値を、セクタ131とEXOR回路134に供給する。

#### 【0060】

一方、FIFOメモリ133は、レジスタ132から新たな値D133（ $i=1$ から $i=d_c$ までの $\Pi \text{sign}(v_i)$ ）がEXOR回路134に供給されるまでの間、符号ビットD121を遅延し、1ビットの値D134としてEXOR回路134に供給する。EXOR回路134は、レジスタ132か

ら供給された値D133と、FIFOメモリ133から供給された値D134との排他的論理和を演算することにより、値D133を、値D134で除算し、1ビットの除算結果を除算値D135として出力する。即ち、EXOR回路134は、検査行列の1行に亘る全ての枝からのメッセージD101の符号ビットD121 ( $\text{sign}(|v_i|)$ ) の乗算値を、求めたい枝からのメッセージD101の符号ビットD121 ( $\text{sign}(|v_i|)$ ) で除算して、その除算値 ( $i = 1$  から  $i = d_c - 1$  までの  $\Pi \text{sign}(|v_i|)$ ) を除算値D135として出力する。

#### 【0061】

チェックノード計算器101では、LUT128から出力された5ビットの演算結果D129を下位5ビットとするとともに、EXOR回路134から出力された1ビットの除算値D135を最上位ビットとする合計6ビットがメッセージD102 (メッセージ $u_j$ ) として出力される。

#### 【0062】

以上のように、チェックノード計算器101では、式(7)の演算が行われ、メッセージ $u_j$ が求められる。

#### 【0063】

なお、図8の検査行列の行の重みの最大は9であるため、即ち、チェックノードに供給されるメッセージの最大数は9であるため、チェックノード計算器101は、9個のメッセージ ( $\phi(|v_i|)$ ) を遅延させるFIFOメモリ127とFIFOメモリ133を有している。行の重みが9未満の行のメッセージを計算するときには、FIFOメモリ127とFIFOメモリ133における遅延量が、その行の重みの値に減らされる。

#### 【0064】

図11は、バリアブルノード演算を1つずつ行う図9のバリアブルノード計算器103の構成例を示している。

#### 【0065】

なお、図11では、各メッセージが符号ビットを合わせて合計6ビット(bit)に量子化されているものとして、バリアブルノード計算器103を表している。また、図11では、図8の検査行列で表わされるLDPC符号のバリアブルノード演算が行われる。さらに、図11のバリアブルノード計算機103には、クロック $ck$ が供給され、クロック $ck$ は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック $ck$ に同期して処理を行う。

#### 【0066】

図11のバリアブルノード計算器103は、制御部105から供給される、例えば、1ビットの制御信号D107に基づき、枝用メモリ102から1つずつ読み込まれるメッセージD103と、受信用メモリ104から読み込まれる受信データD104を用いて、式(1)にしたがって演算を行う。

#### 【0067】

即ち、バリアブルノード計算器103では、検査行列の各行に対応するチェックノードからの6ビットのメッセージD103 (メッセージ $u_j$ ) が1つずつ読み込まれ、そのメッセージD103が、加算器151とFIFOメモリ155に供給される。また、バリアブルノード計算器103では、受信用メモリ104から6ビットの受信データD104が1つずつ読み込まれ、加算器156に供給される。さらに、バリアブルノード計算器103には、制御部105から制御信号D107が供給され、その制御信号D107は、セクタ153に供給される。

#### 【0068】

加算器151は、メッセージD103 (メッセージ $u_j$ ) とレジスタ152に格納されている9ビットの値D151とを加算することにより、メッセージD103を積算し、その結果得られる9ビットの積算値を、レジスタ152に再格納する。なお、検査行列の1列に亘る全ての枝からのメッセージD103が積算された場合、レジスタ152はリセットされる。

#### 【0069】

検査行列の1列に亘るメッセージD103が1つずつ読み込まれ、レジスタ152に1列分のメッセージD103が積算された値が格納された場合、制御部105から供給される制御信号D107は、「0」から「1」に変化する。例えば、列の重みが「5」である場合、制御信号



D107は、1から4クロック目までは「0」となり、5クロック目では「1」となる。

#### 【0070】

制御信号D107が「1」の場合、セクタ153は、レジスタ152に格納されている値、即ち、検査行列の1列に亘る全ての枝からのメッセージD103（メッセージ $u_j$ ）が積算された9ビットの値D151（ $j=1$ から $d_v$ までの $\sum u_j$ ）を選択し、レジスタ154に出力して格納させる。レジスタ154は、格納している値D151を、9ビットの値D152として、セクタ153と加減算器156に供給する。制御信号D107が「0」の場合、セクタ153は、レジスタ154から供給された値D152を選択し、レジスタ154に出力し再格納させる。即ち、検査行列の1列に亘る全ての枝からのメッセージD103（メッセージ $u_j$ ）が積算されるまで、レジスタ154は、前回積算された値を、セクタ153と加減算器156に供給する。

#### 【0071】

一方、FIFOメモリ155は、レジスタ154から新たな値D152（ $j=1$ から $d_v$ までの $\sum u_j$ ）が出力されるまでの間、チェックノードからのメッセージD103を遅延し、6ビットの値D153として加減算器156に供給する。加減算器156は、レジスタ154から供給された値D152から、FIFOメモリ155から供給された値D153を減算する。即ち、加減算器156は、検査行列の1列に亘る全ての枝からのメッセージD103（メッセージ $u_j$ ）の積算値から、求めたい枝からのメッセージ $u_j$ を減算して、その減算値（ $j=1$ から $d_v-1$ までの $\sum u_j$ ）を求める。さらに、加減算器156には、その減算値（ $j=1$ から $d_v-1$ までの $\sum u_j$ ）に、受信用メモリ104から供給された受信データD104を加算して、その結果得られる6ビットの値をメッセージD105（メッセージ $v_i$ ）として出力する。

#### 【0072】

以上のように、バリアブルノード計算器103では、式（1）の演算が行われ、メッセージ $v_i$ が求められる。

#### 【0073】

なお、図8の検査行列の列の重みの最大は5であるため、即ち、バリアブルノードに供給されるメッセージの最大数は5であるため、バリアブルノード計算器103は、5個のメッセージ（ $u_j$ ）を遅延させるFIFOメモリ155を有している。列の重みが5未満の列のメッセージを計算するときには、FIFOメモリ155における遅延量が、その列の重みの値に減らされる。

#### 【0074】

図9の復号装置では、検査行列の重みにしたがって、制御部105から制御信号が与えられる。そして、図9の復号装置によれば、枝用メモリ100および102、並びにチェックノード計算器101およびバリアブルノード計算器103のFIFOメモリ127、133、155の容量さえ足りれば、制御信号のみを変えることで様々な検査行列のLDPC符号を復号することができる。

#### 【0075】

なお、図示しないが、図9の復号装置において、復号の最終段においては、式（1）のバリアブルノード演算の代わりに、式（5）の演算が行われ、その演算結果が、最終的な復号結果として出力される。

#### 【0076】

図9の復号装置を繰り返し用いて、LDPC符号を復号する場合には、チェックノード演算とバリアブルノード演算とが交互に行われる。即ち、図9の復号装置では、チェックノード計算器101によるチェックノード演算の結果を用いて、バリアブルノード計算器103によりバリアブルノード演算が行われ、バリアブルノード計算器103によるバリアブルノード演算の結果を用いて、チェックノード計算器101によりチェックノード演算が行われる。

#### 【0077】

従って、269の枝を有する図8の検査行列を用いた1回の復号に、 $269 \times 2 = 538$ クロック（clock）を必要とする。例えば、50回の繰り返し復号を行うためには、符号長である

90個の符号(受信データ)を1フレームとして、その1フレームを受信する間に、 $538 \times 50 = 26900$ クロック動作することが必要であり、受信周波数の約300 ( $\div 26900/90$ ) 倍の高速動作が必要になる。受信周波数が数十MHzであるとする、GHz以上の速度での動作を要求されることになる

**【0078】**

また、図9の復号装置を、例えば、50台接続して、LDPC符号を復号する場合には、1フレーム(frame)目のバリエブルノード演算を行っている間に、2フレーム目のチェックノード演算を行い、3フレーム目のバリエブルノード演算を行う、というように、複数のバリエブルノード演算とチェックノード演算とを同時に行うことができる。この場合、90個の符号を受信する間に、269個の枝を計算すればよいので、復号装置は、受信周波数の約3 ( $\div 269/90$ ) 倍の周波数で動作すればよいことになり、十分に実現可能である。しかしながら、この場合、回路規模が、単純には、図9の復号装置の50倍になる。

**【0079】**

次に、全ノードの演算を同時に行うことによって復号を行う場合(full parallel decoding)の復号装置の実装法について説明する。

**【0080】**

この実装法については、例えば、非特許文献1に記載されている。

**【0081】**

図12は、図8の検査行列で表現される符号(符号化率2/3、符号長90)を復号する復号装置の一例の構成を示している。

**【0082】**

図12の復号装置では、枝用メモリ202または206から、269個ある枝に対応するメッセージデータを全て同時に読み出し、そのメッセージデータを用いて、269個の枝に対応する新たなメッセージデータを演算する。さらに、その演算の結果求められた新たなメッセージデータが全て同時に後段の枝用メモリ206または202に格納されていく。そして、図12の復号装置を繰り返し用いることで繰り返し復号が実現される。

**【0083】**

図12において、復号装置は、1つの受信用メモリ205、2つの枝入れ替え装置200および203、2つの枝用メモリ202および206、30個のチェックノード計算器201<sub>1</sub>乃至201<sub>30</sub>、90個のバリエブルノード計算器204<sub>1</sub>乃至204<sub>90</sub>からなる。以下、各部について詳細に説明する。

**【0084】**

枝用メモリ206は、前段のバリエブルノード計算器204<sub>1</sub>乃至204<sub>90</sub>からのメッセージD206<sub>1</sub>乃至D206<sub>90</sub>を全て同時に格納し、次の時刻(次のクロックのタイミング)に、メッセージD206<sub>1</sub>乃至D206<sub>90</sub>を、メッセージD207<sub>1</sub>乃至D207<sub>90</sub>として読み出し、次段の枝入れ替え装置200に、メッセージD200(D200<sub>1</sub>乃至D200<sub>90</sub>)として供給する。枝入れ替え装置200は、枝用メモリ206から供給されたメッセージD200<sub>1</sub>乃至D200<sub>90</sub>の順番を、図8の検査行列にしたがって並び替え(入れ替え)、チェックノード計算器201<sub>1</sub>乃至201<sub>30</sub>に、メッセージD201<sub>1</sub>乃至D201<sub>30</sub>として供給する。

**【0085】**

チェックノード計算器201<sub>1</sub>乃至201<sub>30</sub>は、枝入れ替え装置200から供給されるメッセージD201<sub>1</sub>乃至D201<sub>30</sub>を用いて式(7)にしたがって演算を行い、その演算の結果得られるメッセージD202<sub>1</sub>乃至D202<sub>30</sub>を、枝用メモリ202に供給する。

**【0086】**

枝用メモリ202は、前段のチェックノード計算器201<sub>1</sub>乃至201<sub>30</sub>から供給されるメッセージD202<sub>1</sub>乃至D202<sub>30</sub>を全て同時に格納し、次の時刻に、そのすべてのメッセージD202<sub>1</sub>乃至D202<sub>30</sub>を、メッセージD203<sub>1</sub>乃至D203<sub>30</sub>として、次段の枝入れ替え装置203に供給する。

**【0087】**

枝入れ替え装置203は、枝用メモリ202から供給されたメッセージD203<sub>1</sub>乃至D203<sub>30</sub>

の順番を図8の検査行列にしたがって並び替え、バリエブルノード計算器204<sub>1</sub>乃至204<sub>90</sub>に、メッセージD204<sub>1</sub>乃至D204<sub>90</sub>として供給する。

**【0088】**

バリエブルノード計算器204<sub>1</sub>乃至204<sub>90</sub>は、枝入れ替え装置203から供給されるメッセージD204<sub>1</sub>乃至D204<sub>90</sub>と、受信用メモリ205から供給される受信データD205<sub>1</sub>乃至D205<sub>90</sub>を用いて式(1)にしたがって演算を行い、その演算の結果得られるメッセージD206<sub>1</sub>乃至D206<sub>90</sub>を、次段の枝用メモリ206に供給する。

**【0089】**

図13は、チェックノード演算を同時に行う図12のチェックノード計算器201<sub>m</sub> ( $m=1, 2, \dots, 30$ )の構成例を示している。

**【0090】**

図13のチェックノード計算器201<sub>m</sub>では、図10のチェックノード計算器101と同様にして、式(7)のチェックノード演算が行われるが、そのチェックノード演算が、すべての枝について同時に行われる。

**【0091】**

即ち、図13のチェックノード計算器201<sub>m</sub>では、枝入れ替え装置200から供給される図8の検査行列の各列に対応するバリエブルノードからのメッセージD221<sub>1</sub>乃至D221<sub>9</sub> ( $v_i$ ) が全て同時に読み込まれ、それぞれの下位5ビットである絶対値D222<sub>1</sub>乃至D222<sub>9</sub> ( $|v_i|$ ) がLUT221<sub>1</sub>乃至221<sub>9</sub>にそれぞれ供給される。また、メッセージD221<sub>1</sub>乃至D221<sub>9</sub> ( $v_i$ ) の最上位ビットである1ビットの符号ビットD223<sub>1</sub>乃至D223<sub>9</sub>が、EXOR回路226<sub>1</sub>乃至226<sub>9</sub>にそれぞれ供給されるとともに、EXOR回路225に供給される。

**【0092】**

LUT221<sub>1</sub>乃至221<sub>9</sub>は、絶対値D222<sub>1</sub>乃至D222<sub>9</sub> ( $|v_i|$ ) に対して、式(7)における $\phi(|v_i|)$ の演算を行った5ビットの演算結果D224<sub>1</sub>乃至D224<sub>9</sub> ( $\phi(|v_i|)$ ) をそれぞれ読み出し、それぞれを減算器223<sub>1</sub>乃至223<sub>9</sub>に供給する。また、LUT221<sub>1</sub>乃至221<sub>9</sub>は、演算結果D224<sub>1</sub>乃至D224<sub>9</sub> ( $\phi(|v_i|)$ ) を加算器222に供給する。

**【0093】**

加算器222は、演算結果D224<sub>1</sub>乃至D224<sub>9</sub> ( $\phi(|v_i|)$ ) の値の総和(1行分の演算結果の総和)を演算し、9ビットの演算結果D225 ( $i=1$ から9の $\sum \phi(|v_i|)$ ) を、減算器223<sub>1</sub>乃至223<sub>9</sub>に供給する。減算器223<sub>1</sub>乃至223<sub>9</sub>は、演算結果D225から、演算結果D224<sub>1</sub>乃至D224<sub>9</sub> ( $\phi(|v_i|)$ ) をそれぞれ減算し、5ビットの減算値D227<sub>1</sub>乃至D227<sub>9</sub>を、LUT224<sub>1</sub>乃至224<sub>9</sub>に供給する。即ち、減算器223<sub>1</sub>乃至223<sub>9</sub>は、全ての枝からのメッセージ $v_i$ から求められた $\phi(|v_i|)$ の積算値から、求めたい枝からのメッセージ $v_i$ から求められた $\phi(|v_i|)$ を減算して、その減算値D227<sub>1</sub>乃至D227<sub>9</sub> ( $i=1$ から8までの $\sum \phi(|v_i|)$ ) をLUT224<sub>1</sub>乃至224<sub>9</sub>にそれぞれ供給する。LUT224<sub>1</sub>乃至224<sub>9</sub>は、減算値D227<sub>1</sub>乃至D227<sub>9</sub>に対して、式(7)における $\phi^{-1}(\sum \phi(|v_i|))$ の演算を行った5ビットの演算結果D228<sub>1</sub>乃至D228<sub>9</sub>を読み出して出力する。

**【0094】**

一方、EXOR回路225は、全ての符号ビットD223<sub>1</sub>乃至D223<sub>9</sub>の排他的論理和を演算することにより、符号ビットD223<sub>1</sub>乃至D223<sub>9</sub>の乗算を行い、1ビットの乗算値D226 (1行分の符号ビットの乗算値 ( $i=1$ から9までの $\prod \text{sign}(v_i)$ )) をEXOR回路226<sub>1</sub>乃至226<sub>9</sub>にそれぞれ供給する。EXOR回路226<sub>1</sub>乃至226<sub>9</sub>は、乗算値D226と符号ビットD223<sub>1</sub>乃至D223<sub>9</sub>それぞれとの排他的論理を演算することにより、乗算値D226を、符号ビットD223<sub>1</sub>乃至D223<sub>9</sub>それぞれで除算した1ビットの除算値D229<sub>1</sub>乃至D229<sub>9</sub> ( $i=1$ から8までの $\prod \text{sign}(v_i)$ ) を求めて出力する。

**【0095】**

チェックノード計算器201<sub>m</sub>では、LUT224<sub>1</sub>乃至224<sub>9</sub>から出力された5ビットの演算結果D228<sub>1</sub>乃至D228<sub>9</sub>それぞれを下位5ビットとするとともに、EXOR回路226<sub>1</sub>乃至226<sub>9</sub>から出力された除算値D229<sub>1</sub>乃至D229<sub>9</sub>それぞれを最上位ビットとする合計6ビットが、チェックノード演算の結果得られるメッセージD230<sub>1</sub>乃至D230<sub>9</sub>として出力される。

## 【0096】

以上のように、チェックノード計算器 201<sub>m</sub>では、式(7)の演算が行われ、メッセージ  $u_j$  が求められる。

## 【0097】

なお、図13では、各メッセージが符号ビットを合わせて合計6ビットに量子化されているものとして、チェックノード計算器 201<sub>m</sub>を表している。また、図13の回路は1つのチェックノードに相当する。ここで処理の対象としている図8の検査行列については、その行数である30行のチェックノードが存在するため、図12の復号装置は、図13に示したようなチェックノード計算器 201<sub>m</sub>を30個有している。

## 【0098】

ここで、図13のチェックノード計算器 201<sub>m</sub>では、9個のメッセージを同時に計算することができる。そして、ここで処理の対象としている図8の検査行列の行の重みは、第1行が8で、第2乃至第30行が9であるため、即ち、チェックノードに供給されるメッセージの数が、8のケースが1つと、9のケースが29あるため、チェックノード計算器 201<sub>1</sub>は、図13の回路と同様の8つのメッセージを同時に計算することができる回路構成となっており、残りのチェックノード計算器 201<sub>2</sub>乃至201<sub>30</sub>は、図13の回路と同一構成となっている。

## 【0099】

図14は、バリアブルノード演算を同時に行う図12のバリアブルノード計算器 204<sub>p</sub> ( $p = 1, 2, \dots, 90$ )の構成例を示している。

## 【0100】

図14のバリアブルノード計算器 204<sub>p</sub>では、図11のバリアブルノード計算器 103と同様にして、式(1)のバリアブルノード演算が行われるが、そのバリアブルノード演算が、すべての枝について同時に行われる。

## 【0101】

即ち、図14のバリアブルノード計算器 204<sub>p</sub>では、枝入れ替え装置 203から供給される、検査行列の各行に対応するチェックノードからの6ビットのメッセージ  $D251_1$ 乃至  $D251_5$  (メッセージ  $u_j$ ) が全て同時に読み込まれ、それぞれ加算器 252<sub>1</sub>乃至252<sub>5</sub>に供給されるとともに、加算器 251に供給される。また、バリアブルノード計算器 204<sub>p</sub>には、受信用メモリ 205から受信データ  $D271$ が供給され、その受信データ  $D271$ は、加減算器 252<sub>1</sub>乃至252<sub>5</sub>に供給される。

## 【0102】

加算器 251は、全てのメッセージ  $D251_1$ 乃至  $D251_5$  (メッセージ  $u_j$ ) を積算し、9ビットの積算値  $D252$  (1列分のメッセージの総和値 ( $j = 1$  から 5 までの  $\sum u_j$ )) を加減算器 252<sub>1</sub>乃至252<sub>5</sub>に供給する。加減算器 252<sub>1</sub>乃至252<sub>5</sub>は、加算値  $D252$  から、メッセージ  $D251_1$ 乃至  $D251_5$  (メッセージ  $u_j$ ) をそれぞれ減算する。即ち、加減算器 252<sub>1</sub>乃至252<sub>5</sub>は、全ての枝からのメッセージ  $u_j$  の積算値  $D252$  から、求めたい枝からのメッセージ  $D251_1$ 乃至  $D251_5$  (メッセージ  $u_j$ ) をそれぞれ減算して、その減算値 ( $j = 1$  から 4 までの  $\sum u_j$ ) を求める。

## 【0103】

さらに、加減算器 252<sub>1</sub>乃至252<sub>5</sub>は、減算値 ( $j = 1$  から 4 までの  $\sum u_j$ ) に、受信データ  $D271$  ( $u_{0i}$ ) を加算して、6ビットの加算値  $D253_1$ 乃至  $D253_5$  を、バリアブルノード演算の結果として出力する。

## 【0104】

以上のように、バリアブルノード計算器 204<sub>p</sub>では、式(1)の演算が行われ、メッセージ  $v_i$  が求められる。

## 【0105】

なお、図14では、各メッセージが符号ビットを合わせて合計6ビットに量子化されているものとして、バリアブルノード計算器 204<sub>p</sub>を表している。また、図14の回路は1つのバリアブルノードに相当する。ここで処理の対象としている図8の検査行列につい

ては、その列数である 90 列のバリアブルノードが存在するから、図 12 の復号装置は、図 14 に示したような回路を 90 個有している。

#### 【0106】

ここで、図 14 のバリアブルノード計算器 204<sub>p</sub>では、5 個のメッセージを同時に計算することができる。そして、ここで処理の対象としている図 8 の検査行列は、重みが 5, 3, 2, 1 の列が、それぞれ、15 列、45 列、29 列、1 列あるので、バリアブルノード計算器 204<sub>1</sub>乃至 204<sub>90</sub>のうちの 15 個は、図 14 の回路と同一構成となっており、残りの 45 個、29 個、1 個は、図 14 の回路と同様の 3, 2, 1 つのメッセージをそれぞれ同時に計算することができる回路構成となっている。

#### 【0107】

なお、図示しないが、図 12 の復号装置においても、図 9 における場合と同様に、復号の最終段においては、式 (1) のバリアブルノード演算の代わりに、式 (5) の演算が行われ、その演算結果が最終的な復号結果として出力される。

#### 【0108】

図 12 の復号装置によれば、269 個ある枝に対応するメッセージすべてを 1 クロックで同時に計算することができる。

#### 【0109】

図 12 の復号装置を繰り返し用いて復号する場合には、チェックノード演算とバリアブルノード演算とを交互に行い、1 回の復号を 2 クロックで行うことができる。従って、例えば、50 回の復号を行うためには、符号長が 90 個の符号を 1 フレームとする受信データを受信する間に  $2 \times 50 = 100$  クロック動作すれば良いことになり、ほぼ受信周波数と同一の動作周波数でよいことになる。一般的に、LDPC 符号は、符号長が数千から数万と大きいことから、図 12 の復号装置を用いれば、復号回数を極めて多くすることができ、誤り訂正性能の向上を期待することができる。

#### 【0110】

しかしながら、図 12 の復号装置は、タナグラフのすべての枝に対応するメッセージの演算を、並列で行うため、回路規模が、符号長に比例して大きくなる。また、図 12 の復号装置を、ある符号長の、ある符号化率の、ある検査行列を持つ LDPC 符号の復号を行う装置として構成した場合、その復号装置において、他の符号長や、他の符号化率、他の検査行列を持つ LDPC 符号の復号を行うことは困難となる。即ち、図 12 の復号装置は、図 9 の復号装置のように、制御信号を変えるだけでは、様々な符号を復号することに対処することが困難であり、符号依存性が高い。

#### 【0111】

図 9 および図 12 の復号装置の他に、一つでも全てでもなく、4 つずつのメッセージの計算を同時に行う実装法について、例えば、非特許文献 2 に述べられているが、この場合、メモリの異なるアドレスからの同時読み出し、もしくは同時書き込みを避けることが一般的には容易でなく、メモリアクセス制御が困難であるという問題がある。

#### 【0112】

また、サムプロダクトアルゴリズムを近似して実装する方法なども提案されているが、この方法では、性能の劣化を招いてしまう。

【非特許文献 1】 C. Howland and A. Blanksby, "Parallel Decoding Architectures for Low Density Parity Check Codes", Symposium on Circuits and Systems, 2001

【非特許文献 2】 E. Yeo, P. Pakzad, B. Nikolic and V. Anantharam, "VLSI Architectures for iterative Decoders in Magnetic Recording Channels", IEEE Transactions on Magnetics, Vol. 37, No. 2, March 2001

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0113】

サムプロダクトアルゴリズムをハードウェアに実装する場合には、上述したように、枝

に対応するメッセージの演算（チェックノード演算とビットノード(bit node)計算）を、1つずつシリアル(serial)に行う方法、すべて並列（フルパラレル(full parallel)）に行う方法、幾つかずつ並列（パラレル(parallel)）に行う方法がある。

**【0114】**

しかしながら、枝に対応するメッセージの演算を1つずつ行う方法では、高い動作周波数が必要となる。そこで、スループット(throughput)を上げる方法として、装置を、パイプライン(pipeline)化する方法があるが、この場合、回路規模、特にメモリ（の容量）が大きくなってしまう。

**【0115】**

また、メッセージの演算を全て並列に行う方法では、ロジック(logic)の回路規模が大きくなるとともに、符号依存性が高くなる。

**【0116】**

さらに、メッセージの演算を、幾つかずつ並列に行う方法では、メモリアクセスの制御が難しくなる。

**【0117】**

本発明は、このような状況に鑑みてなされたものであり、ロジック、メモリ共に回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行うことができるようにするものである。

**【課題を解決するための手段】****【0118】**

本発明の復号装置は、 $P \times P$ の単位行列、その単位行列のコンポーネントである1のうちの1個以上が0になった行列である準単位行列、単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または $P \times P$ の0行列を構成行列として、LDPC符号の検査行列が、複数の構成行列の組合せで表される場合において、LDPC符号の復号のためのP個のチェックノードの演算を同時に行う第1の演算手段と、LDPC符号の復号のためのP個のバリエアブルノードの演算を同時に行う第2の演算手段とを備えることを特徴とする。

**【0119】**

第1の演算手段は、チェックノードの演算を行うP個のチェックノード計算器を有し、第2の演算手段は、バリエアブルノードの演算を行うP個のバリエアブルノード計算器を有するようにすることができる。

**【0120】**

P個のチェックノードの演算、またはP個のバリエアブルノードの演算の結果得られるP個の枝に対応するメッセージデータを同時に読み書きするメッセージ記憶手段をさらに備えるようにすることができる。

**【0121】**

メッセージ記憶手段は、チェックノードの演算時に読み出される枝に対応するメッセージデータを、検査行列の1を行方向に詰めるように格納するようにすることができる。

**【0122】**

メッセージ記憶手段は、バリエアブルノード演算時に読み出される枝に対応するメッセージデータを、検査行列の1を列方向に詰めるように格納するようにすることができる。

**【0123】**

メッセージ記憶手段は、検査行列を表す構成行列のうちの、重みが2以上の構成行列について、その構成行列を、重みが1の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列に属するP個の枝に対応するメッセージを、同一のアドレスに格納するようにすることができる。

**【0124】**

メッセージ記憶手段は、行数/P個のFIFOと、列数/P個のFIFOとで構成され、行数/P個のFIFOと列数/P個のFIFOは、それぞれ、検査行列の行と列の重みに対応するワード数を有するようにすることができる。

**【0125】**

メッセージ記憶手段は、RAM(Random Access Memory)で構成され、RAMは、メッセージデータを、読み出される順番に詰めて格納し、格納位置順に読み出すようにすることができる。

**【0126】**

LDPC符号の受信情報を格納するとともに、P個の受信情報を同時に読み出す受信情報記憶手段をさらに備えるようにすることができる。

**【0127】**

受信情報記憶手段は、受信情報を、バリエブルノードの演算に必要なとなる順番に読み出すことができるように格納するようにすることができる。

**【0128】**

P個のチェックノードの演算、またはP個のバリエブルノードの演算の結果得られるメッセージを並べ替える並べ替え手段をさらに備えるようにすることができる。

**【0129】**

並べ替え手段は、バレルシフタで構成されるようにすることができる。

**【0130】**

第1の演算手段と第2の演算手段は、P個の枝に対応するメッセージを求めるようにすることができる。

**【0131】**

第1の演算手段は、P個のチェックノードの演算とP個のバリエブルノードの演算の一部とを行い、第2の演算手段は、P個のバリエブルノードの演算の他の一部を行うようにすることができる。

**【0132】**

第1の演算手段は、P個のチェックノードの演算とP個のバリエブルノードの演算の一部を行うP個の計算器を有し、第2の演算手段は、P個のバリエブルノードの演算の他の一部を行うP個の計算器を有するようにすることができる。

**【0133】**

第1の演算手段がP個のチェックノードの演算とP個のバリエブルノードの演算の一部を行うことにより得られるP個の枝に対応する第1の復号途中結果を同時に読み書きする第1の復号途中結果記憶手段をさらに備えるようにすることができる。

**【0134】**

第1の復号途中記憶手段は、P個のバリエブルノードの演算の他の一部を行う時に読み出される枝に対応する第1の復号途中結果を、検査行列の1を行方向に詰めるように格納するようにすることができる。

**【0135】**

第1の復号途中結果記憶手段は、2個のシングルポートRAM(Random Access Memory)であるようにすることができる。

**【0136】**

2個のシングルポートRAMは、第1の復号途中結果をP個ずつ交互に格納するようにすることができる。

**【0137】**

2個のシングルポートRAM(Random Access Memory)は、それぞれ同一のアドレスに格納している第1の復号途中結果を読み出すようにすることができる。

**【0138】**

第1の復号途中結果記憶手段は、検査行列を表す構成行列のうちの、重みが2以上の構成行列について、その構成行列を、重みが1の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列に属するP個の枝に対応する第1の復号途中結果を、同一のアドレスに格納するようにすることができる。

**【0139】**

第2の演算手段がP個のバリエブルノードの演算の他の一部を行うことにより得られるP個の枝に対応する第2の復号途中結果を同時に読み書きする第2の復号途中結果記憶手段をさらに備えるようにすることができる。

【0140】

LDPC符号の受信情報を格納するとともに、P個の受信情報を同時に読み出す受信情報記憶手段をさらに備えるようにすることができる。

【0141】

受信情報記憶手段は、受信情報を、P個のバリエブルノードの演算の他の一部の演算に必要な順番に読み出すことができるように格納するようにすることができる。

【0142】

第1の演算手段がP個のチェックノードの演算とP個のバリエブルノードの演算の一部を行うことにより得られる第1の復号途中結果、または第2の演算手段がP個のバリエブルノードの演算の他の一部を行うことにより得られる第2の復号途中結果を並べ替える並べ替え手段をさらに備えるようにすることができる。

【0143】

並べ替え手段は、バレルシフタで構成されるようにすることができる。

【0144】

第1の演算手段は、P個のチェックノードの演算の一部を行い、第2の演算手段は、P個のチェックノードの演算の他の一部と、P個のバリエブルノードの演算とを行うようにすることができる。

【0145】

第1の演算手段は、P個のチェックノードの演算の一部を行うP個の計算器を有し、第2の演算手段は、P個のチェックノードの演算の他の一部と、P個のバリエブルノードの演算を行うP個の計算器を有することができる。

【0146】

第1の演算手段がP個のチェックノードの演算の一部を行うことにより得られるP個の枝に対応する第1の復号途中結果を同時に読み書きする第1の復号途中結果記憶手段をさらに備えるようにすることができる。

【0147】

第2の演算手段がP個のチェックノードの演算の他の一部と、P個のバリエブルノードの演算を行うことにより得られるP個の枝に対応する第2の復号途中結果を同時に読み書きする第2の復号途中結果記憶手段をさらに備えるようにすることができる。

【0148】

第2の復号途中結果記憶手段は、P個のチェックノードの演算の他の一部と、P個のバリエブルノードの演算を行う時に読み出される枝に対応する第2の復号途中結果を、検査行列の1を列方向に詰めるように格納するようにすることができる。

【0149】

第2の復号途中結果記憶手段は、2個のシングルポートRAM(Random Access Memory)であるようにすることができる。

【0150】

2個のシングルポートRAMは、第2の復号途中結果をP個ずつ交互に格納することができる。

【0151】

2個のシングルポートRAM(Random Access Memory)は、それぞれ同一のアドレスに格納している第2の復号途中結果を読み出すようにすることができる。

【0152】

第2の復号途中結果記憶手段は、検査行列を表す構成行列のうちの、重みが2以上の構成行列について、その構成行列を、重みが1の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列に属するP個の枝に対応する第2の復号途中結果を、同一のアドレスに格納するようにする



ことができる。

**【0153】**

LDPC符号の受信情報を格納するとともに、P個の受信情報を同時に読み出す受信情報記憶手段をさらに備えるようにすることができる。

**【0154】**

請求項36に記載の復号装置であって、受信情報記憶手段は、受信情報を、P個のチェックノードの演算の他の一部と、P個のバリエーションノードの演算に必要となる順番に読み出すことができるように格納するようにすることができる。

**【0155】**

第1の演算手段がP個のチェックノードの演算の一部を行うことにより得られる第1の復号途中結果、または第2の演算がP個のチェックノードの演算の他の一部と、P個のバリエーションノードの演算を行うことにより得られる第2の復号途中結果を並べ替える並べ替え手段をさらに備えるようにすることができる。

**【0156】**

並べ替え手段は、バレルシフタで構成されるようにすることができる。

**【0157】**

本発明の復号方法は、 $P \times P$ の単位行列、その単位行列のコンポーネントである1のうちの1個以上が0になった行列である準単位行列、単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または $P \times P$ の0行列を構成行列として、LDPC符号の検査行列が、複数の構成行列の組合せで表される場合において、LDPC符号の復号のためのP個のチェックノードの演算を同時に行う第1の演算ステップと、LDPC符号の復号のためのP個のバリエーションノードの演算を同時に行う第2の演算ステップとを含むことを特徴とする。

**【0158】**

本発明のプログラムは、LDPC符号の復号のためのP個のチェックノードの演算を同時に行う第1の演算ステップと、LDPC符号の復号のためのP個のバリエーションノードの演算を同時に行う第2の演算ステップとを含むことを特徴とする。

**【0159】**

本発明においては、 $P \times P$ の単位行列、その単位行列のコンポーネントである1のうちの1個以上が0になった行列である準単位行列、単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または $P \times P$ の0行列を構成行列として、LDPC符号の検査行列が、複数の構成行列の組合せで表される場合において、LDPC符号の復号のためのP個のチェックノードの演算が同時に行われ、LDPC符号の復号のためのP個のバリエーションノードの演算が同時に行われる。

**【発明の効果】****【0160】**

本発明によれば、回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行うことが可能となる。また、LDPC符号の復号装置の回路規模を小さくすることができるので、コストを削減し、装置の消費電力を削減することができる。

**【発明を実施するための最良の形態】****【0161】**

以下に本発明の実施の形態を説明するが、請求項に記載の構成要件と、発明の実施の形態における具体例との対応関係を例示すると、次のようになる。この記載は、請求項に記載されている発明をサポートする具体例が、発明の実施の形態に記載されていることを確認するためのものである。従って、発明の実施の形態中には記載されているが、構成要件に対応するものとして、ここには記載されていない具体例があったとしても、そのことは、その具体例が、その構成要件に対応するものではないことを意味するものではない。逆に、具体例が構成要件に対応するものとしてここに記載されていたとしても、そのことは

、その具体例が、その構成要件以外の構成要件には対応しないものであることを意味するものでもない。

**【0162】**

さらに、この記載は、発明の実施の形態に記載されている具体例に対応する発明が、請求項に全て記載されていることを意味するものではない。換言すれば、この記載は、発明の実施の形態に記載されている具体例に対応する発明であって、この出願の請求項には記載されていない発明の存在、すなわち、将来、分割出願されたり、補正により追加される発明の存在を否定するものではない。

**【0163】**

請求項1に記載の復号装置は、LDPC(Low Density Parity Check)符号の復号装置であって、 $P \times P$ の単位行列、その単位行列のコンポーネントである1のうちの1個以上が0になった行列である準単位行列、前記単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、前記単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または $P \times P$ の0行列を構成行列として、前記LDPC符号の検査行列が、複数の前記構成行列の組合せで表される場合において、前記LDPC符号の復号のためのP個のチェックノードの演算を同時に行う第1の演算手段(例えば、図16のチェックノード計算部313)と、前記LDPC符号の復号のためのP個のバリエブルノードの演算を同時に行う第2の演算手段(例えば、図16のバリエブルノード計算部319)とを備えることを特徴とする。

**【0164】**

請求項2に記載の復号装置は、請求項1に記載の復号装置であって、前記第1の演算手段は、チェックノードの演算を行うP個のチェックノード計算器(例えば、図16のチェックノード計算器313<sub>1</sub>)を有し、前記第2の演算手段は、バリエブルノードの演算を行うP個のバリエブルノード計算器(例えば、図16のバリエブルノード計算器319<sub>1</sub>)を有することを特徴とする。

**【0165】**

請求項3に記載の復号装置は、請求項1に記載の復号装置であって、前記P個のチェックノードの演算、または前記P個のバリエブルノードの演算の結果得られるP個の枝に対応するメッセージデータを同時に読み書きするメッセージ記憶手段(例えば、図16の枝データ格納メモリ311または316)をさらに備えることを特徴とする。

**【0166】**

請求項9に記載の復号装置は、請求項1に記載の復号装置であって、LDPC符号の受信情報を格納するとともに、P個の前記受信情報を同時に読み出す受信情報記憶手段(例えば、図16の受信データ用メモリ318)をさらに備えることを特徴とする。

**【0167】**

請求項11に記載の復号装置は、請求項1に記載の復号装置であって、前記P個のチェックノードの演算、または前記P個のバリエブルノードの演算の結果得られるメッセージを並べ替える並べ替え手段(例えば、図16のサイクリックシフト回路314または320)をさらに備えることを特徴とする。

**【0168】**

請求項15に記載の復号装置は、請求項14に記載の復号装置であって、前記第1の演算手段は、前記P個のチェックノードの演算と前記P個のバリエブルノードの演算の一部を行うP個の計算器(例えば、計算器412<sub>1</sub>)を有し、前記第2の演算手段は、前記P個のバリエブルノードの演算の他の一部を行うP個の計算器(例えば、計算器415<sub>1</sub>)を有することを特徴とする。

**【0169】**

請求項16に記載の復号装置は、請求項14に記載の復号装置であって、前記第1の演算手段が前記P個のチェックノードの演算と前記P個のバリエブルノードの演算の一部を行うことにより得られるP個の枝に対応する第1の復号途中結果を同時に読み書きする第1の復号途中結果記憶手段(例えば、図18の復号途中結果格納用メモリ413)をさらに

備えることを特徴とする。

【0170】

請求項18に記載の復号装置は、請求項16に記載の復号装置であって、前記第1の復号途中結果記憶手段は、2個のシングルポートRAM(Random Access Memory) (例えば、図23の復号途中結果格納用RAM502または503) であるようにすることができる。

【0171】

請求項22に記載の復号装置は、請求項14に記載の復号装置であって、前記第2の演算手段が前記P個のバリエブルノードの演算の他の一部を行うことにより得られるP個の枝に対応する前記第2の復号途中結果を同時に読み書きする第2の復号途中結果記憶手段 (例えば、図18の復号途中結果格納用メモリ410) をさらに備えることを特徴とする。

【0172】

請求項23に記載の復号装置は、請求項14に記載の復号装置であって、LDPC符号の受信情報を格納するとともに、P個の前記受信情報を同時に読み出す受信情報記憶手段 (例えば、図18の受信用メモリ416) をさらに備えることを特徴とする。

【0173】

請求項25に記載の復号装置は、請求項14に記載の復号装置であって、前記第1の演算手段が前記P個のチェックノードの演算と前記P個のバリエブルノードの演算の一部を行うことにより得られる第1の復号途中結果、または前記第2の演算手段が前記P個のバリエブルノードの演算の他の一部を行うことにより得られる第2の復号途中結果を並べ替える並べ替え手段 (例えば、図18のサイクリックシフト回路411または414) をさらに備えることを特徴とする。

【0174】

請求項28に記載の復号装置は、請求項27に記載の復号装置であって、前記第1の演算手段は、前記P個のチェックノードの演算の一部を行うP個の計算器 (例えば、図26の計算器612<sub>1</sub>) を有し、前記第2の演算手段は、前記P個のチェックノードの演算の他の一部と、前記P個のバリエブルノードの演算を行うP個の計算器 (例えば、図26の計算器615<sub>1</sub>) を有することを特徴とする。

【0175】

請求項29に記載の復号装置は、請求項27に記載の復号装置であって、前記第1の演算手段が前記P個のチェックノードの演算の一部を行うことにより得られるP個の枝に対応する第1の復号途中結果を同時に読み書きする第1の復号途中結果記憶手段 (例えば、図26の復号途中結果格納用メモリ613) をさらに備えることを特徴とする。

【0176】

請求項30に記載の復号装置は、請求項27に記載の復号装置であって、前記第2の演算手段が前記P個のチェックノードの演算の他の一部と、前記P個のバリエブルノードの演算を行うことにより得られるP個の枝に対応する第2の復号途中結果を同時に読み書きする第2の復号途中結果記憶手段 (例えば、図26の復号途中結果格納用メモリ610) をさらに備えることを特徴とする。

【0177】

請求項32に記載の復号装置は、請求項30に記載の復号装置であって、前記第2の復号途中結果記憶手段は、2個のシングルポートRAM(Random Access Memory) (例えば、図31の復号途中結果格納用RAM702と703) であることを特徴とする。

【0178】

請求項36に記載の復号装置は、請求項27に記載の復号装置であって、LDPC符号の受信情報を格納するとともに、P個の前記受信情報を同時に読み出す受信情報記憶手段 (例えば、図26の受信用メモリ616) をさらに備えることを特徴とする。

【0179】

請求項38に記載の復号装置は、請求項27に記載の復号装置であって、前記第1の演算手段が前記P個のチェックノードの演算の一部を行うことにより得られる第1の復号途中結果、または前記第2の演算が前記P個のチェックノードの演算の他の一部と、前記P個

のバリアブルノードの演算を行うことにより得られる第2の復号途中結果を並べ替える並べ替え手段(例えば、図26のサイクリックシフト回路611または614)をさらに備えることを特徴とする。

#### 【0180】

請求項40に記載の復号方法は、LDPC(Low Density Parity Check)符号の復号装置の復号方法であって、 $P \times P$ の単位行列、その単位行列のコンポーネントである1のうちの1個以上が0になった行列である準単位行列、前記単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、前記単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または $P \times P$ の0行列を構成行列として、前記LDPC符号の検査行列が、複数の前記構成行列の組合せで表される場合において、前記LDPC符号の復号のためのP個のチェックノードの演算を同時に行う第1の演算ステップ(例えば、図17のステップS36)と、前記LDPC符号の復号のためのP個のバリアブルノードの演算を同時に行う第2の演算ステップ(例えば、図17のステップS31)とを含むことを特徴とする。

#### 【0181】

請求項41に記載のプログラムの各ステップの具体例も、請求項40に記載の復号方法の各ステップの発明の実施の形態における具体例と同様である。

#### 【0182】

以下、本発明を適用した具体的な実施の形態について、図面を参照しながら詳細に説明する。

#### 【0183】

図15は、 $5 \times 5$ の行列の単位に間隔を空けた $30 \times 90$ の検査行列の例を示している。なお、図15の検査行列自体は、図8に示した検査行列と同一である。

#### 【0184】

図15においては、検査行列は、 $5 \times 5$ の単位行列、その単位行列の1のうちの1個以上が0になった行列(以下、適宜、準単位行列という)、単位行列または準単位行列をサイクリックシフト(cyclic shift)した行列(以下、適宜、シフト行列という)、単位行列、準単位行列、またはシフト行列のうちの2以上(複数)の和(以下、適宜、和行列という)、 $5 \times 5$ の0行列の組合せで表わされている。なお、図15の検査行列で表現されるLDPC符号は、符号化率 $2/3$ 、符号長90である。

#### 【0185】

図15の検査行列は、 $5 \times 5$ の単位行列、準単位行列、シフト行列、和行列、0行列で構成されているといえることができる。そこで、検査行列を構成する、これらの $5 \times 5$ の行列を、以下、適宜、構成行列という。

#### 【0186】

図16は、図15の検査行列で表現されるLDPC符号を復号する復号装置の一実施の形態の構成例を示している。

#### 【0187】

図16の復号装置300は、スイッチ310および315、6つのFIFO311<sub>1</sub>乃至311<sub>6</sub>からなる枝データ格納メモリ311、セクタ312、5つのチェックノード計算器313<sub>1</sub>乃至313<sub>5</sub>からなるチェックノード計算部313、2つのサイクリックシフト回路314および320、18個のFIFO316<sub>1</sub>乃至316<sub>18</sub>からなる枝データ格納メモリ316、セクタ317、受信情報を格納する受信データ用メモリ318、バリアブルノード計算部319、制御部321から構成される。

#### 【0188】

この復号装置300の各部について詳細に説明する前に、まず、枝データ格納メモリ311と316へのデータの格納方法について説明する。

#### 【0189】

枝データ格納メモリ311は、検査行列の行数30を構成行列の行数5で除算した数である6つのFIFO311<sub>1</sub>乃至311<sub>6</sub>から構成されている。FIFO311<sub>y</sub> ( $y=1, 2, \dots$ )

・ ・ ・ , 6) は、構成行列の行数および列数である 5 つの枝に対応するメッセージを同時に読み出しもしくは書き込むことができるようになっており、その長さ (段数) は、検査行列の行方向の 1 の数 (ハミング重み) の最大数である 9 になっている。

#### 【0190】

FIFO3 1 1<sub>1</sub>には、図 15 の検査行列の第 1 行目から第 5 行目までの 1 の位置に対応するデータが、各行共に横方向 (列方向) に詰めた形に (0 を無視した形で) 格納される。すなわち、第  $j$  行第  $i$  列を、 $(j, i)$  と表すこととすると、FIFO3 1 1<sub>1</sub>の第 1 の要素 (第 1 段) には、検査行列の  $(1, 1)$  から  $(5, 5)$  の  $5 \times 5$  の単位行列の 1 の位置に対応するデータが格納される。第 2 の要素には、検査行列の構成行列である  $(1, 21)$  から  $(5, 25)$  のシフト行列 ( $5 \times 5$  の単位行列を右方向に 3 つだけサイクリックシフトしたシフト行列) の 1 の位置に対応するデータが格納される。第 3 から第 8 の要素も同様に検査行列の構成行列と対応するデータが格納される。そして、第 9 の要素には、検査行列の  $(1, 86)$  から  $(5, 90)$  のシフト行列 ( $5 \times 5$  の単位行列のうちの 1 行目の 1 を 0 に置き換えて 1 つだけ左にサイクリックシフトしたシフト行列) の 1 の位置に対応するデータが格納される。ここで、検査行列の  $(1, 86)$  から  $(5, 90)$  のシフト行列においては、1 行目に 1 がいないため、FIFO3 1 1<sub>1</sub>の 1 行目のみ要素数は 8、残りの行は要素数が 9 となる。

#### 【0191】

FIFO3 1 1<sub>2</sub>には、図 15 の検査行列の第 6 行目から第 10 行目までの 1 の位置に対応するデータが格納される。すなわち、FIFO3 1 1<sub>2</sub>の第 1 の要素には、検査行列の  $(6, 1)$  から  $(10, 5)$  の和行列 ( $5 \times 5$  の単位行列を右に 1 つだけサイクリックシフトした第 1 のシフト行列と、右に 2 つだけサイクリックシフトした第 2 のシフト行列の和である和行列) を構成する第 1 のシフト行列の 1 の位置に対応するデータが格納される。また、第 2 の要素には、検査行列の  $(6, 1)$  から  $(10, 5)$  の和行列を構成する第 2 のシフト行列の 1 の位置に対応するデータが格納される。

#### 【0192】

即ち、重みが 2 以上の構成行列については、その構成行列を、重みが 1 である  $P \times P$  の単位行列、そのコンポーネントである 1 のうち 1 個以上が 0 になった準単位行列、または単位行列もしくは準単位行列をサイクリックシフトしたシフト行列のうちの複数の和の形で表現したときの、その重みが 1 の単位行列、準単位行列、またはシフト行列の 1 の位置に対応するデータ (単位行列、準単位行列、またはシフト行列に属する枝に対応するメッセージ) は、同一アドレス (FIFO3 1 1<sub>1</sub>乃至 3 1 1<sub>6</sub>のうちの同一のFIFO) に格納される。

#### 【0193】

以下、第 3 から第 9 の要素についても、検査行列に対応づけてデータが格納される。FIFO3 1 1<sub>2</sub>は全行共に要素数は 9 となる。

#### 【0194】

FIFO3 1 1<sub>3</sub>乃至 3 1 1<sub>6</sub>も同様に検査行列に対応づけてデータを格納し、各FIFO3 1 1<sub>3</sub>乃至 3 1 1<sub>6</sub>それぞれの長さは 9 である。

#### 【0195】

枝データ格納メモリ 3 1 6 は、検査行列の列数 90 を、構成行列の列数である 5 で割った 18 個の FIFO3 1 6<sub>1</sub>乃至 3 1 6<sub>18</sub>から構成されている。FIFO3 1 6<sub>x</sub> ( $x = 1, 2, \dots, 18$ ) は、構成行列の行数および列数である 5 つの枝に対応するメッセージを同時に読み出しもしくは書き込むことができるようになっている。

#### 【0196】

FIFO3 1 6<sub>1</sub>には、図 15 の検査行列の第 1 列目から第 5 列目までの 1 の位置に対応するデータが、各列共に縦方向 (行方向) に詰めた形に (0 を無視した形で) 格納される。すなわち、FIFO3 1 6<sub>1</sub>の第 1 の要素 (第 1 段) には、検査行列の  $(1, 1)$  から  $(5, 5)$  の  $5 \times 5$  の単位行列の 1 の位置に対応するデータが格納される。第 2 の要素には、検査行列の  $(6, 1)$  から  $(10, 5)$  の和行列 ( $5 \times 5$  の単位行列を右に 1 つだけサイクリックシフトした第 1 のシフト行列と、右に 2 つだけサイクリックシフトした第 2 のシフト行列との和である和行列) を構成する第 1 のシフト行列の 1 の位置に対応するデータが格納される。また、第

3の要素には、検査行列の(6,1)から(10,5)の和行列を構成する第2のシフト行列の1の位置に対応するデータが格納される。

**【0197】**

即ち、重みが2以上の構成行列については、その構成行列を、重みが1である $P \times P$ の単位行列、そのコンポーネントである1のうち1個以上が0になった準単位行列、または単位行列もしくは準単位行列をサイクリックシフトしたシフト行列のうちの複数の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列の1の位置に対応するデータ(単位行列、準単位行列、またはシフト行列に属する枝に対応するメッセージ)は、同一アドレス(FIFO316<sub>1</sub>乃至316<sub>18</sub>のうちの同一のFIFO)に格納される。

。

**【0198】**

以下、第4および第5の要素についても、検査行列に対応づけて、データが格納される。このFIFO316<sub>1</sub>の要素数(段数)は、検査行列の第1列から第5列における行方向の1の数(ハミング重み)の最大数である5になっている。

**【0199】**

FIFO316<sub>2</sub>と316<sub>3</sub>も同様に検査行列に対応づけてデータを格納し、それぞれの長さ(段数)は、5である。FIFO316<sub>4</sub>乃至316<sub>12</sub>も同様に検査行列に対応づけてデータを格納し、それぞれの長さは3である。FIFO316<sub>13</sub>乃至316<sub>18</sub>も同様に検査行列に対応づけてデータを格納し、それぞれの長さは2である。但し、FIFO316<sub>18</sub>の第1の要素は、検査行列の(1,86)から(5,90)に相当し、第5列目(検査行列の(1,90)から(5,90))に1がないため、データは格納されない。

**【0200】**

以下、図16の復号装置300の各部の動作について詳細に説明する。

**【0201】**

スイッチ310には、サイクリックシフト回路320から5つのメッセージ(データ)D319が供給されるとともに、制御部321から検査行列のどの行に属するかの情報(Matrixデータ)を表す制御信号D320が供給される。制御信号D320にしたがって、5つのメッセージ(データ)D319を格納するFIFOを、FIFO311<sub>1</sub>乃至311<sub>6</sub>の中から選択し、選択したFIFOに5つのメッセージデータD319をまとめて順番に供給していく。

**【0202】**

枝データ格納メモリ311は、6つのFIFO311<sub>1</sub>乃至311<sub>6</sub>からなる。枝データ格納メモリ311のFIFO311<sub>1</sub>乃至311<sub>6</sub>には、スイッチ310から、5つのメッセージD319がまとめて順番に供給され、FIFO311<sub>1</sub>乃至311<sub>6</sub>は、5つのメッセージD319をまとめて順番に(同時に)格納していく。また、枝データ格納メモリ311は、データを読み出す際には、FIFO311<sub>1</sub>から5つのメッセージ(データ)D311<sub>1</sub>を順番に読み出し、次段のセクタ312に供給する。枝データ格納メモリ311は、FIFO311<sub>1</sub>からのメッセージD311<sub>1</sub>の読み出しの終了後、FIFO311<sub>2</sub>乃至311<sub>6</sub>からも、順番に、メッセージD311<sub>1</sub>乃至D311<sub>6</sub>をそれぞれ読み出し、セクタ312に供給する。

**【0203】**

セクタ312には、制御部321から、FIFO311<sub>1</sub>乃至311<sub>6</sub>のうち、メッセージデータを読み出すFIFO(現在データが読み出されているFIFO)の選択を表す選択信号D321が供給されるとともに、枝データ格納メモリ311から5つのメッセージ(データ)D311<sub>1</sub>乃至D311<sub>6</sub>が供給される。セクタ312は、選択信号D321にしたがって、FIFO311<sub>1</sub>乃至311<sub>6</sub>のうちの、現在データが読み出されているFIFOを選択し、その選択したFIFOから供給された5つのメッセージデータを、メッセージD312として、チェックノード計算部313に供給する。

**【0204】**

チェックノード計算部313は、5つのチェックノード計算器313<sub>1</sub>乃至313<sub>5</sub>からなる。チェックノード計算部313には、セクタ312を介して5つのメッセージD312が供給され、そのメッセージD312が、チェックノード計算器313<sub>1</sub>乃至313<sub>5</sub>のそれぞれ

れに1つずつ供給される。また、チェックノード計算部313には、制御部321から制御信号D322が供給され、その制御信号D322が、チェックノード計算器313<sub>1</sub>乃至313<sub>5</sub>に供給される。チェックノード計算器313<sub>1</sub>乃至313<sub>5</sub>は、メッセージD312を用いて、式(7)にしたがって同時に演算を行い、その演算の結果、5個の枝に対応するメッセージD313を求める。チェックノード計算部313は、チェックノード計算器313<sub>1</sub>乃至313<sub>5</sub>による演算の結果得られる5つのメッセージD313をサイクリックシフト回路314に供給する。

#### 【0205】

ここで、制御部321からチェックノード計算部313に供給される制御信号D322は、図10の制御信号D106に対応するものであり、チェックノード計算部313<sub>1</sub>乃至313<sub>5</sub>それぞれは、図10に示したチェックノード計算器101と同様に構成される。

#### 【0206】

サイクリックシフト回路314には、チェックノード計算部313で計算された5つのメッセージD313が供給されるとともに、制御部321から、そのメッセージD313に対応する枝が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報(Matrixデータ)を表す制御信号D323が供給される。サイクリックシフト回路314は、制御信号D323を元に、5つのメッセージD313を並べ替えるサイクリックシフトを行い、その結果をメッセージD314として、スイッチ315に供給する。

#### 【0207】

スイッチ315には、サイクリックシフト回路314から供給される5つのメッセージ(データ)D314が検査行列のどの列に属するかの情報を表す制御信号D324が供給されるとともに、サイクリックシフト回路314から、メッセージD314が供給される。スイッチ315は、制御信号D324にしたがって、メッセージD314を格納するFIFOを、FIFO316<sub>1</sub>乃至316<sub>18</sub>の中から選択し、選択したFIFOに5つのメッセージD314をまとめて順番に供給していく。

#### 【0208】

枝データ格納メモリ316は、18個のFIFO316<sub>1</sub>乃至316<sub>18</sub>からなる。枝データ格納メモリ316のFIFO316<sub>1</sub>乃至316<sub>18</sub>には、スイッチ315から5つのメッセージD314がまとめて順番に(同時に)供給され、FIFO316<sub>1</sub>乃至316<sub>18</sub>は、その5つのメッセージD314をまとめて順番に格納していく。また、枝データ格納メモリ316は、データを読み出す際には、FIFO316<sub>1</sub>から5つのメッセージD315<sub>1</sub>を順番に読み出し、次段のセクタ317に供給する。枝データ格納メモリ316は、FIFO316<sub>1</sub>からのデータの読み出しの終了後、FIFO316<sub>2</sub>乃至316<sub>18</sub>からも、順番に、メッセージD315<sub>2</sub>乃至D315<sub>18</sub>を読み出し、セクタ317に供給する。

#### 【0209】

セクタ317には、制御部321からFIFO316<sub>1</sub>乃至316<sub>18</sub>のうち、メッセージデータを読み出すFIFO(現在データが読み出されているFIFO)の選択を表す選択信号D325が供給されるとともに、枝データ格納メモリ316からメッセージデータD315<sub>1</sub>乃至D315<sub>18</sub>が供給される。セクタ317は、選択信号D325にしたがって、FIFO316<sub>1</sub>乃至316<sub>18</sub>のうちの、現在データが読み出されているFIFOを選択し、その選択したFIFOから供給される5つのメッセージデータを、メッセージD316として、バリアブルノード計算部319と上述した式(5)の演算を行う不図示のブロックに供給する。

#### 【0210】

一方、受信データ用メモリ318は、通信路を通して受信した受信信号から、受信LLR(対数尤度比)を計算しており、その計算した受信LLRを5つまとめて(同時に)受信データD317(LDPC符号)としてバリアブルノード計算部319と、式(5)の演算を行う不図示のブロックに供給する。なお、受信データ用メモリ318は、バリアブルノード計算部319のバリアブルノード演算に必要となる順番に、受信データD317を読み出す。

#### 【0211】

バリアブルノード計算部319は、5つのバリアブルノード計算器319<sub>1</sub>乃至319<sub>5</sub>

からなる。バリアブルノード計算部 319 には、セクタ 317 を介して 5 つのメッセージ D316 が供給され、そのメッセージ D316 が、バリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> のそれぞれに 1 つずつ供給される。また、バリアブルノード計算部 319 には、受信データ用メモリ 318 から 5 つの受信データ D317 が供給され、その受信データ D317 が、バリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> のそれぞれに 1 つずつ供給される。さらに、バリアブルノード計算部 319 には、制御部 321 から制御信号 D326 が供給され、その制御信号 D326 がバリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> に供給される。

#### 【0212】

バリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> は、メッセージ D316 と、受信データ D317 を用いて、式 (1) にしたがって同時に演算を行い、その演算の結果、5 個の枝に対応するメッセージ D318 を求める。バリアブルノード計算部 319 は、バリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> の結果得られる 5 つのメッセージ D318 を、サイクリックシフト回路 320 に供給する。

#### 【0213】

ここで、制御部 521 からバリアブルノード計算部 319 に供給される制御信号 D326 は、図 11 の制御信号 D107 に対応するものであり、バリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> それぞれは、図 11 のバリアブルノード計算器 103 と同様に構成される。

#### 【0214】

サイクリックシフト回路 320 には、バリアブルノード計算部 319 から 5 つのメッセージ D318 が供給されるとともに、制御部 321 から、そのメッセージ D318 に対応する枝が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報を表す制御信号 D327 が供給される。サイクリックシフト回路 320 は、制御信号 D327 を元に、メッセージ D327 を並べ替えるサイクリックシフトを行い、その結果をメッセージ D319 として、スイッチ 310 に供給する。

#### 【0215】

なお、制御部 321 は、制御信号 D320 をスイッチ 310 に、選択信号 D321 をセクタ 312 に供給することにより、それぞれを制御する。また制御部 321 は、制御信号 D322 をチェックノード計算部 313 に、制御信号 D323 をサイクリックシフト回路 314 に、制御信号 D324 をスイッチ 315 に供給することにより、それぞれを制御する。さらに、制御部 321 は、選択信号 D325 をセクタ 317、制御信号 D326 をバリアブルノード計算部 319 に、制御信号 D327 をサイクリックシフト回路 320 に供給することにより、それぞれを制御する。

#### 【0216】

以上の動作を 1 巡することで、LDPC 符号の 1 回の復号を行うことができる。図 16 の復号装置 300 は、所定の回数だけ LDPC 符号を復号した後、図示しないが、式 (5) にしたがって最終的な復号結果を求めて出力する。

#### 【0217】

なお、枝データ (枝に対応するメッセージ) が欠けている箇所に関しては、メモリ格納時 (枝データ格納メモリ 311 と 316 へのデータ格納時) には、何のメッセージも格納せず、また、ノード演算時 (チェックノード計算部 313 でのチェックノード演算時とバリアブルノード計算部 319 でのバリアブルノード演算時) にも何の演算も行わない。

#### 【0218】

図 17 は、図 16 の復号装置 300 の復号処理を説明するフローチャートである。この処理は、例えば、受信データ用メモリ 318 に復号すべき受信データが格納されたとき、開始される。

#### 【0219】

ステップ S31 において、バリアブルノード計算部 319 は、バリアブルノード演算を行う。

#### 【0220】

具体的には、バリアブルノード計算部 319 には、セクタ 317 を介して、5 つのメ



ッセージD316 (メッセージ $u_j$ ) が供給される。即ち、枝データ格納メモリ 316 は、後述するステップ S39 で格納された FIFO 316<sub>1</sub> から 5 つのメッセージ D316<sub>1</sub> を順番に読み出し、その後、FIFO 316<sub>2</sub> 乃至 316<sub>18</sub> からも、順番に、メッセージ D316<sub>2</sub> 乃至 D316<sub>18</sub> を読み出して、セクタ 317 に供給する。

#### 【0221】

セクタ 317 には、制御部 321 から FIFO 316<sub>1</sub> 乃至 316<sub>18</sub> のうち、メッセージ (データ) を読み出す FIFO (現在データが読み出されている FIFO) の選択を表す選択信号 D307 が供給されるとともに、枝データ格納メモリ 316 からメッセージデータ D316<sub>1</sub> 乃至 D316<sub>18</sub> が供給される。セクタ 317 は、選択信号 D307 にしたがって、FIFO 316<sub>1</sub> 乃至 316<sub>18</sub> のうちの、現在データが読み出されている FIFO を選択し、その選択した FIFO から供給される 5 つのメッセージデータを、メッセージ D316 として、バリアブルノード計算部 319 に供給する。

#### 【0222】

なお、受信データ用メモリ 306 から供給された受信データ D309 に対して、まだチェックノード演算が行われておらず、枝データ格納メモリ 316 にメッセージ D304 が格納されていない場合、バリアブルノード計算部 319 は、バリアブルノード演算に用いるメッセージ $u_j$  を初期値に設定する。

#### 【0223】

また、バリアブルノード計算部 319 には、受信データ用メモリ 318 から 5 つの受信データ D309 (受信値 $u_{0i}$ ) が供給され、その受信データ D309 が、バリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> のそれぞれに 1 つずつ供給される。さらに、バリアブルノード計算部 319 には、制御部 321 から制御信号 D315 が供給され、その制御信号 D315 がバリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> に供給される。

#### 【0224】

バリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> は、メッセージ D316 と、受信データ D309 を用いて、制御信号 D315 に基づいて、式 (1) にしたがって同時に演算を行い、その演算の結果 5 つのメッセージ D319 を求める。

#### 【0225】

即ち、制御部 321 がバリアブルノード計算部 319 に供給する制御信号 D315 は、前述の図 11 で説明した制御信号 D107 に対応するものであり、バリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> は、制御信号 D309 にしたがって、セクタ 317 を介して、枝データ格納メモリ 316 から必要なメッセージ D314 (D316) を、それぞれ 1 つずつ読み出すとともに、受信データ用メモリ 318 から供給された 5 つの受信データ D309 を、それぞれ 1 つずつ読み出して、バリアブルノード演算を行い、その演算の結果 5 つのメッセージ D319 を同時に求める。

#### 【0226】

ステップ S31 の処理後は、ステップ S32 に進み、バリアブルノード計算部 319 は、バリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> のバリアブルノード演算の結果得られる 5 つのメッセージ D319 (メッセージ $v_i$ ) をサイクリックシフト回路 320 に供給し、ステップ S33 に進む。

#### 【0227】

ステップ S33 において、サイクリックシフト回路 320 は、バリアブルノード計算部 319 から供給された 5 つのメッセージ D318 を、サイクリックシフトする (並べ替える)。

#### 【0228】

具体的には、サイクリックシフト回路 320 には、バリアブルノード計算部 319 からメッセージ D318 が供給されるとともに、制御部 321 から、そのメッセージ D318 に対応する枝が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報 (Matrix データ) を表す制御信号 D327 が供給される。サイクリックシフト回路 320 は、制御信号 D327 を元に、5 つのメッセージ D327 をサイクリックシフトし、その結果

をメッセージD319として、スイッチ310に供給する。

#### 【0229】

ステップS33の処理後は、ステップS34に進み、スイッチ310は、サイクリックシフト回路320から供給される5つのメッセージD319を枝データ格納メモリ311に供給する。

#### 【0230】

具体的には、スイッチ310には、サイクリックシフト回路320からメッセージ(データ)D304が供給されるとともに、そのメッセージD304が検査行列のどの行に属するかの情報を表す制御信号D312が供給される。スイッチ310は、制御信号D312にしたがって、メッセージD304を格納するFIFOを、枝データ格納メモリ311のFIFO300<sub>1</sub>乃至300<sub>6</sub>の中から選択し、選択したFIFOに5つのメッセージデータD304をまとめて順番に供給していく。

#### 【0231】

そして、枝データ格納メモリ311のFIFO300<sub>1</sub>乃至300<sub>18</sub>は、スイッチ310から供給された5つのメッセージデータD304をまとめて順番に格納していく。

#### 【0232】

ステップS34の処理後は、ステップS35に進み、制御部321は、バリエابلノード計算部319により、全枝数のメッセージが演算されたかどうかを判定し、全枝数のメッセージが演算されていないと判定した場合、ステップS31に戻り、上述した処理を繰り返す。

#### 【0233】

一方、ステップS35において、バリエابلノード計算部319は、全枝数のメッセージが演算されたと判定した場合、ステップS36に進み、チェックノード計算部313は、チェックノード演算を行う。

#### 【0234】

具体的には、チェックノード計算部313には、セクタ312を介して、5つのメッセージD302が供給される。即ち、枝データ格納メモリ311は、ステップS34で格納されたFIFO311<sub>1</sub>から5つのメッセージD311<sub>1</sub>(メッセージv<sub>i</sub>)を順番に読み出し、その後、FIFO311<sub>2</sub>乃至311<sub>6</sub>からも、順番に、メッセージデータD311<sub>2</sub>乃至D311<sub>6</sub>を読み出し、セクタ312に供給する。

#### 【0235】

セクタ312には、制御部321からFIFO311<sub>1</sub>乃至311<sub>6</sub>のうち、メッセージデータを読み出すFIFO(現在データが読み出されているFIFO)の選択を表す選択信号D321が供給されるとともに、枝データ格納メモリ311からメッセージデータD311<sub>1</sub>乃至D311<sub>6</sub>が供給される。セクタ301は、選択信号D321にしたがって、FIFO311<sub>1</sub>乃至311<sub>6</sub>のうちの、現在データが読み出されているFIFOを選択し、その選択したFIFOから供給される5つのメッセージデータを、メッセージD311として、チェックノード計算部313に供給する。

#### 【0236】

また、チェックノード計算部313には、制御部321から制御信号D322が供給される。チェックノード計算部313のチェックノード計算器313<sub>1</sub>乃至313<sub>5</sub>は、制御信号D322に基づいて、メッセージD302を用いて、上述した式(7)にしたがって同時にチェックノード演算を行い、その演算結果である5つのメッセージD303(メッセージu<sub>j</sub>)を求める。

#### 【0237】

即ち、制御部321がチェックノード計算部313に供給する制御信号D322は、前述の図10で説明した制御信号D106に対応するものであり、チェックノード計算器313<sub>1</sub>乃至313<sub>5</sub>は、制御信号D322にしたがい、セクタ312を介して、枝データ格納メモリ311から必要なメッセージD311(D312)を、それぞれ1つずつ読み出しながら、チェックノード演算を行い、その演算の結果5つのメッセージD313を同時に求める。

## 【0238】

ステップS37の処理後は、ステップS38に進み、チェックノード計算部313は、チェックノードの演算の結果得られる5つのメッセージD313をサイクリックシフト回路314に出力して、ステップS38に進む。

## 【0239】

ステップS38において、サイクリックシフト回路314は、チェックノード計算部313から供給された5つのメッセージD313を、サイクリックシフトする。

## 【0240】

具体的には、サイクリックシフト回路314には、チェックノード計算部313からメッセージD313が供給されるとともに、制御部321から、そのメッセージD313に対応する枝が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報 (Matrixデータ) を表す制御信号D314が供給される。サイクリックシフト回路314は、制御信号D314を元に、5つのメッセージD313をサイクリックシフトし、その結果をメッセージD304として、スイッチ315に供給する。

## 【0241】

ステップS38の処理後は、ステップS39に進み、スイッチ315は、サイクリックシフト回路314から供給される5つのメッセージD304を枝データ格納メモリ316に格納する。

## 【0242】

具体的には、スイッチ316には、サイクリックシフト回路314から、5つのメッセージ (データ) D304が供給されるとともに、そのメッセージ (データ) D304が検査行列のどの列に属するかの情報を表す制御信号D324が供給される。スイッチ316は、制御信号D324にしたがって、メッセージD304を格納するFIFOを、枝データ格納メモリ316のFIFO316<sub>1</sub>乃至316<sub>18</sub>の中から選び、選んだFIFOに5つのメッセージデータD304をまとめて順番に供給していく。

## 【0243】

そして、枝データ格納メモリ316のFIFO316<sub>1</sub>乃至316<sub>18</sub>は、スイッチ316から供給された5つのメッセージデータD304をまとめて順番に格納していく。

## 【0244】

ステップS39の処理後は、ステップS40に進み、制御部321は、チェックノード計算部313により、全枝数のメッセージが演算されたかどうかを判定し、全枝数のメッセージが演算されていないと判定した場合、ステップS36に戻り、上述した処理を繰り返す。

## 【0245】

一方、ステップS40において、制御部321は、チェックノード計算部313により、全枝数のメッセージが演算されたと判定した場合、処理を終了する。

## 【0246】

なお、復号装置300は、復号回数だけ図17の復号処理を繰り返し行ない、チェックノード計算部313が、最後のチェックノード演算を行った場合、チェックノード演算の結果得られるメッセージD304が、枝データ格納メモリ316から、セクタ317を介して、上述した式(5)の演算を行う不図示のブロックに供給される。不図示のブロックは、さらに受信データ用メモリ306から受信データD309が供給され、不図示のブロックは、メッセージD304と受信データD309を用いて、式(5)の演算を行い、その演算結果を最終的な復号結果として出力する。

## 【0247】

上記説明には、枝データ格納にFIFOを用いたが (枝データ格納メモリ311と316をFIFOで構成するようにしたが)、FIFOの代わりにRAMを用いても構わない。その場合、RAMには、P個の枝情報 (枝に対応するメッセージ) を同時に読み出すことの出来るビット幅と、枝総数/Pのワード(word)数が必要となる。さらに、RAMへの書き込みは、検査行列の情報から、書き込もうとしているデータが次に読み出される際に何番目に読み出されるか

を求め、その位置に書き込む。また、RAMからの読み出しの際には、アドレスの先頭から順次データを読み出す。即ち、RAMは、メッセージデータを読み出される順番に詰めて格納し、格納位置順に読み出す。FIFOの代わりにRAMを用いると、セクタ312および317は不要になる。

#### 【0248】

なお、FIFOやRAMの物理的なビット幅が足りない場合には、複数のRAMを用いて同じ制御信号を与えることで、論理的に1つのRAMとみなすことができる。

#### 【0249】

ところで、図16の復号装置300では、チェックノード演算の結果得られるメッセージ $u_j$ を用いて、バリアブルノード演算が行われ、その演算の結果得られるメッセージ $v_i$ を用いて、チェックノード演算が行われるため、チェックノード演算の結果得られる枝に対応するメッセージ $u_j$ とバリアブルノード演算の結果得られる枝に対応するメッセージ $v_i$ をすべて格納する枝データ格納メモリ311と枝データ格納メモリ316が必要である。即ち、復号装置では、検査行列Hの“1”の数の2倍のメッセージを格納するために必要な容量のメモリが必要である。

#### 【0250】

そこで、復号装置の回路規模さらに小さくするため、図16の復号装置300に比べて、さらにメモリの容量を減らした復号装置を以下に示す。

#### 【0251】

図18は、本発明を適用した図15の検査行列で表現されるLDPC符号を復号する復号装置の他の一実施の形態の構成例を示すブロック図である。

#### 【0252】

図18の復号装置400では、図16の枝データ格納メモリ311が、枝データ格納メモリ311に比べて容量の小さい復号途中結果格納用メモリ410になっている。

#### 【0253】

復号装置400は、復号途中結果格納用メモリ410、サイクリックシフト回路411、5つの計算器412<sub>1</sub>乃至計算器412<sub>5</sub>からなる計算部412、復号途中結果格納用メモリ413、サイクリックシフト回路414、5つの計算器415<sub>1</sub>乃至計算器415<sub>5</sub>からなる計算部415、受信用メモリ416、および制御部417から構成される。

#### 【0254】

ここで、図19乃至図22を用いて、図18の計算部412の計算器412<sub>1</sub>乃至計算器412<sub>5</sub>、および計算部415の計算器415<sub>1</sub>乃至計算器415<sub>5</sub>と図10のチェックノード計算器101と図11のバリアブルノード計算器103との関係について説明する。

#### 【0255】

図19と図20は、前述の図10のチェックノード計算器101と図11のバリアブルノード計算器103とそれぞれ同一の図である。また、図21は、計算部412<sub>k</sub> ( $k=1, 2, \dots, 5$ ) の構成例を示しており、図22は、計算部415<sub>k</sub> ( $k=1, 2, \dots, 5$ ) の構成例を示している。

#### 【0256】

図18の復号装置400では、計算器412<sub>k</sub>がチェックノード演算を行い、計算部415<sub>k</sub>が、バリアブルノード演算をおこなうのではなく、計算器412<sub>k</sub>がチェックノード演算とバリアブルノード演算の一部を、計算器415<sub>k</sub>がバリアブルノード演算の他の一部を行う。

#### 【0257】

即ち、図21の計算器412<sub>k</sub>は、ブロックA'とブロックB'から構成されている。ブロックA'は、図19のチェックノード計算器101のチェックノード演算を行うブロックAと同様に構成されている。また、ブロックB'は、図20のバリアブルノード計算器103の一部である、検査行列の各列の全ての枝に対応するメッセージ $u_j$ の積算値から、求めたい枝に対応するメッセージ $u_j$ を減算するブロックBと同様に構成されている。一方

、図22の計算器415<sub>k</sub>は、ブロックC'から構成されている。ブロックC'は、図20のバリアブルノード計算器103の他の一部である、検査行列の各列の枝に対応するメッセージ $u_j$ を積算し、その積算値に受信値 $u_{0i}$ を加算するブロックCと同様に構成されている。

#### 【0258】

そして、図21の計算器412<sub>k</sub>は、ブロックAとブロックBによる演算の結果、即ち、チェックノード演算とバリアブルノード演算の一部を行った復号途中結果 $u_j$ を復号途中結果格納用メモリ413に供給し、図22の計算器415<sub>k</sub>は、バリアブルノード演算の他の一部を行った復号途中結果 $v$ を復号途中結果格納用メモリ410に供給する。

#### 【0259】

従って、図18の復号装置400は、計算器412<sub>k</sub>の演算と計算器415<sub>k</sub>の演算とを交互に行うことにより、チェックノード演算とバリアブルノード演算を行い、復号を行うことができる。

#### 【0260】

なお、図22の計算器412<sub>k</sub>では、復号途中結果格納用メモリ413に格納されている求めたい枝に対応する復号途中結果 $u_j$ を用いて、ブロックBで、計算器415<sub>k</sub>の演算の結果得られる復号途中結果 $v$ から、求めたい枝に対応する復号途中結果 $u_j$ を減算するので、図20のFIFOメモリ155が必要ない。

#### 【0261】

次に、計算器412<sub>k</sub>で行われる演算と、計算器415<sub>k</sub>で行われる演算について、式を用いて説明する。

#### 【0262】

具体的には、計算部412は、上述した式(7)と、以下に表す式(8)にしたがう第1の演算を行い、その第1の演算の結果である復号途中結果 $u_j$ を復号途中結果格納用メモリ410に供給して格納させる。計算部415は、上述した式(5)にしたがう第2の演算を行い、その第2の演算の結果である復号途中結果 $v$ を復号途中結果格納用メモリ410に供給して格納させる。

#### 【0263】

#### 【数8】

$$v_i = v - u_{dv}$$

... (8)

#### 【0264】

なお、式(8)の $u_{dv}$ は、検査行列Hのi列のメッセージを求めようとする枝からのチェックノード演算の途中結果(ここでは、チェックノード演算結果そのもの)を表している。即ち、 $u_{dv}$ は、求めたい枝に対応する復号途中結果である。

#### 【0265】

即ち、上述した式(5)にしたがう第2の演算の結果得られる復号途中結果 $v$ は、受信値 $u_{0i}$ と検査行列Hのi列の各行の1に対応するすべての枝からのチェックノード演算の復号途中結果 $u_j$ とを加算したものである。上述した式(7)に用いられる値 $v_i$ は、式(5)にしたがう第2の演算の結果得られる復号途中結果 $v$ から、検査行列Hのi列の、各行の1に対応する枝からのチェックノード演算の復号途中結果 $u_j$ のうち、メッセージを求めようとする枝からのチェックノード演算の復号途中結果 $u_{dv}$ を引いた値となる。つまり、式(7)の演算に用いられる値 $v_i$ を求める式(1)の演算は、上述した式(5)と式(8)を組み合わせた演算である。

#### 【0266】

従って、復号装置400では、計算部412による式(7)および式(8)にしたがう第1の演算と、計算部415による式(5)にしたがう第2の演算とが交互に行われ、計算部415が、最後の第2の演算の結果を復号結果として出力することにより、LDPC符号の繰り返し復号を行うことができる。

## 【0267】

なお、ここでは、式(7)と式(8)にしたがう第1の演算結果を、復号途中結果 $u_j$ を復号途中結果 $u_j$ と記載するが、この復号途中結果 $u_j$ は、式(7)のチェックノード演算結果 $u_j$ に等しい。

## 【0268】

また、第2の演算により求められる式(5)の $v$ は、式(1)のバリアブルノード演算結果 $v_i$ に対して、メッセージを求めようとする枝からのチェックノード演算結果 $u_j$ を加算したものであるから、検査行列 $H$ の1列(1つのバリアブルノード)に対して、1つだけ求められる。

## 【0269】

復号装置400では、計算部412が、計算部415による第2の演算の結果である検査行列 $H$ の列に対応する復号途中結果 $v$ (第2の復号途中結果)を用いて、第1の演算を行い、その演算の結果得られる検査行列 $H$ の $i$ 列の、各行の1に対応する枝のメッセージ(各チェックノードが各枝に出力するメッセージ)の枝からのチェックノード演算の復号途中結果 $u_j$ (第1の復号途中結果)を復号途中結果格納用メモリ413に格納する。従って、復号途中結果格納用メモリ413の容量は、チェックノード演算の結果を格納する図16の枝データ格納メモリ316と同様に、検査行列の1の数(全枝数)とメッセージの量子化ビット数とを乗算した値となる。一方、計算部415は、計算部412による第1の演算の結果である検査行列 $H$ の $i$ 列の、各行の“1”に対応する復号途中結果 $u_j$ と受信値 $u_{oi}$ を用いて、第2の演算を行い、その演算の結果得られる $i$ 列に対応する復号途中結果 $v$ を復号途中結果格納用メモリ410に格納する。従って、復号途中結果格納用メモリ410に必要な容量は、検査行列の“1”の数より少ない検査行列の列数、即ち、LDPC符号の符号長と復号途中結果 $v$ の量子化ビット数とを乗算した値となる。

## 【0270】

従って、検査行列 $H$ における1が疎らなLDPC符号を復号する復号装置400では、図16の枝データ格納メモリ311に比べて、復号途中結果格納用メモリ410のメモリの容量を削減することができ、これにより、復号装置400の回路規模を小さくすることができる。

## 【0271】

さらに、復号装置400では、計算部415が、式(5)にしたがう第2の演算を行うので、復号装置400は、図16の復号装置300において最終的な復号結果を演算する式(5)の演算を行う不図示のブロックを有する必要がなく、図16の復号装置300に比べて、図18の復号装置の回路規模を小さくすることができる。

## 【0272】

以下、図18の復号装置400の各部の動作について詳細に説明する。

## 【0273】

復号途中結果格納用メモリ410には、計算部415から、計算部415による第2の演算の結果である検査行列の5つの列に対応する5つの復号途中結果D415が供給され、復号途中結果格納用メモリ410は、計算部415から供給された5つの復号途中結果D415を、第1アドレスから順に格納(記憶)する。

## 【0274】

即ち、復号途中結果格納用メモリ410の第1アドレスには、検査行列の列に対応する復号途中結果のうち、第1列目から第5列目の復号途中結果 $v$ が格納される。そして、同様に、第2アドレスには、第6列目から第10列目の復号途中結果 $v$ が格納され、第3アドレスには、第11列目から第15列目の復号途中結果が格納される。以後、同様に、第16列目から第90列目までの復号途中結果 $v$ が、5個ずつ、第4アドレスから第18アドレスまで格納され、計90個の復号途中結果 $v$ が復号途中結果格納用メモリ410に格納される。従って、復号途中結果格納用メモリ410のワード(word)数は、図15の検査行列 $H$ の列数(LDPC符号の符号長)である90を、同時に読み書きする復号途中結果の数である5で割り算した18となる。

## 【0275】

また、復号途中結果格納用メモリ410は、既に格納してある復号途中結果D415から、後段の計算部412が求めようとする復号途中結果 $u_j$ の対応する検査行列Hの行において“1”になっている復号途中結果 $v$ を5つ同時に読み出し、復号途中結果D410として、サイクリックシフト回路411に供給する。

## 【0276】

なお、復号途中結果格納用メモリ410は、例えば、5つの復号途中結果を同時に読み書き可能なシングルポートRAMで構成される。また、復号途中結果格納用メモリ410には、計算部415の第2の演算により演算された列に対応する復号途中結果 $v$ が格納されるので、復号途中結果格納用メモリ410に格納されるデータ量、即ち、復号途中結果格納用メモリ410に必要とされる記憶容量は、復号途中結果の量子化ビット数と、検査行列Hの列数(LDPC符号の符号長)との乗算値である。

## 【0277】

サイクリックシフト回路411には、復号途中結果格納用メモリ410から5つの復号途中結果D410が供給されるとともに、制御部417から、その復号途中結果D410に対応する検査行列の1が、検査行列において元となる単位行列などを幾つサイクリックシフトであるかの情報(Matrixデータ)を表す制御信号D619が供給される。サイクリックシフト回路611は、制御信号D619を元に、5つの復号結果D410を並べ替えるサイクリックシフトを行い、その結果を復号途中結果D411として、計算部412に供給する。

## 【0278】

計算部412は、5つの計算器412<sub>1</sub>乃至412<sub>5</sub>からなる。計算部412には、サイクリックシフト回路411から、計算部415による第2の演算の結果得られた5つの復号途中結果D411(第2の復号途中結果) $v$ が供給されるとともに、復号途中結果格納用メモリ413から、前回、計算器412<sub>1</sub>乃至412<sub>5</sub>による第1の演算の結果得られた5つの復号途中結果D413(第1の復号途中結果) $u_j$ が供給され、その5つの復号途中結果D411と5つの復号途中結果D413が、計算器412<sub>1</sub>乃至412<sub>5</sub>にそれぞれ供給される。また、計算部412には、制御部417から制御信号D419が供給され、その制御信号D419が、計算器412<sub>1</sub>乃至412<sub>5</sub>に供給される。なお、制御信号D419は、5つの計算器412<sub>1</sub>乃至412<sub>5</sub>に共通の信号である。

## 【0279】

計算器412<sub>1</sub>乃至412<sub>5</sub>は、それぞれ復号途中結果D411と復号途中結果D413を用いて、式(7)と式(8)にしたがって第1の演算を行い、復号途中結果D412( $v_i$ )を求める。計算部412は、計算器412<sub>1</sub>乃至412<sub>5</sub>による演算の結果得られる検査行列の5つの1に対応する5つの復号途中結果D412を復号途中結果格納用メモリ413に供給する。

## 【0280】

復号途中結果格納用メモリ413は、例えば、5つの復号途中結果を同時に読み書き可能な、2つのシングルポートRAMから構成される。復号途中結果格納用メモリ413には、計算部412から5つの復号途中結果D412が供給されるとともに、制御部417から復号途中結果413の読み書きを制御する制御信号D420が供給される。

## 【0281】

復号途中結果格納用メモリ413は、制御信号D420に基づいて、計算部412から供給される5つの復号途中結果D412をまとめて格納すると同時に、既に格納してある5つの復号途中結果D412を読み出し、復号途中結果D413として、計算部412とサイクリックシフト回路414に供給する。即ち、復号途中結果格納用メモリ413は、計算部412とサイクリックシフト回路414に供給する復号途中結果D413の読み出しと、計算部412から供給される復号途中結果D412の書き込みとを、同時に行う。

## 【0282】

なお、復号途中結果格納用メモリ413には、計算部412の第1の演算により演算された検査行列Hの $i$ 列の、各行の1に対応する枝からのチェックノード演算の復号途中結果 $u_j$ が格納されるので、復号途中結果格納用メモリ413に格納されるデータ量、即ち、

復号途中結果格納用メモリ413に必要とされる記憶容量は、復号途中結果の量子化ビット数と、検査行列の1の数との乗算値となる。

**【0283】**

サイクリックシフト回路414には、復号途中結果格納用メモリ413から5つの復号途中結果D413（復号途中結果 $u_j$ ）が供給されるとともに、制御部417から、その復号途中結果D413に対応する検査行列の1が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報（Matrixデータ）を表す制御信号D421が供給される。サイクリックシフト回路414は、制御信号D421を元に、5つの復号途中結果D413を並べ替えるサイクリックシフトを行い、その結果を復号途中結果D414として、計算部415に供給する。

**【0284】**

計算部415は、5つの計算器415<sub>1</sub>乃至415<sub>5</sub>からなる。計算部415には、サイクリックシフト回路414から5つの復号途中結果D414が供給され、その復号途中結果D414が、計算器415<sub>1</sub>乃至415<sub>5</sub>のそれぞれに供給される。また、計算部415には、受信用メモリ417から5つの受信データD417（LDPC符号）が供給され、その受信データD417が、計算器415<sub>1</sub>乃至415<sub>5</sub>のそれぞれに供給される。さらに、計算部417には、制御部417から制御信号D422が供給され、その制御信号D422が計算器415<sub>1</sub>乃至415<sub>5</sub>に供給される。なお、制御信号D422は、5つの計算器417<sub>1</sub>乃至417<sub>5</sub>に共通の信号である。

**【0285】**

計算器415<sub>1</sub>乃至415<sub>5</sub>は、それぞれ復号途中結果D414と受信データD417とを用いて、式（5）にしたがって、それぞれ第2の演算を行い、復号途中結果D415を求める。計算部415は、計算器415<sub>1</sub>乃至415<sub>5</sub>の第2の演算の結果得られる5つの復号途中結果D415(v)を、復号途中結果格納用メモリ410に供給する。また、計算部415は、いま行う演算が最後の第2の演算である場合、その演算の結果得られる5つの復号途中結果D415を、最終的な復号結果として出力する。

**【0286】**

受信用メモリ416は、通信路を通して受信した受信値（符号ビット）D416から計算した符号ビットの0らしさの値である受信LLR（対数尤度比）を、受信データD417として格納する。

**【0287】**

即ち、受信用メモリ416の第1のアドレスには、検査行列の列に対応する受信データD417のうち、検査行列の第1列目から第5列目までに対応する受信データD417が格納される。そして、第2のアドレスには、検査行列の第6列目から第10列目までに対応する受信データD417が格納され、第3アドレスには、検査行列の第11列目から第16列目までに対応する受信データD417が格納される。以後、同様に、第4アドレスから第18アドレスまでに、検査行列の第17列目から第90列目までに対応する受信データD417が、5つずつ格納される。

**【0288】**

そして、受信用メモリ416は、既に格納している受信データD417を、バリアブルノード演算に必要となる順番に5つずつ読み出し、計算部415に供給する。

**【0289】**

なお、受信用メモリ416は、例えば、5つの受信データを同時に読み書き可能なシングルポートRAMから構成される。また、受信用メモリ416に格納されるデータ量、即ち、受信用メモリ315に必要とされる記憶容量は、LDPC符号の符号長と、受信データの量子化ビット数との乗算値である。さらに、受信用メモリ416のワード（word）数は、LDPC符号の符号長、即ち、検査行列の列数である90を、同時に読み出す受信データD417の数である5で割り算した値の18である。

**【0290】**

制御部417は、制御信号D418をサイクリックシフト回路411に、制御信号D419を計



算部 412 に供給することにより、それぞれを制御する。また、制御部 417 は、制御信号 D420 を復号途中結果格納用メモリ 413 に、制御信号 D421 をサイクリックシフト回路 414 に、制御信号 D421 を計算部 415 にそれぞれ供給することにより、それぞれを制御する。

#### 【0291】

復号途中結果格納用メモリ 410、サイクリックシフト回路 411、計算部 412、復号途中結果格納用メモリ 413、サイクリックシフト回路 414、計算部 415 の順で、データが一巡することで、復号装置 400 は、1 回の復号を行うことができる。復号装置 400 では、所定の回数だけ繰り返して復号が行われた後、計算部 415 による第 2 の演算の結果である復号途中結果 D415 が、最終的な復号結果として出力される。

#### 【0292】

図 21 は、図 18 の計算部 412 の計算器 412<sub>1</sub> の構成例を示すブロック図である。

#### 【0293】

なお、図 21 では、計算器 412<sub>1</sub> について説明するが、計算器 412<sub>2</sub> 乃至計算器 412<sub>5</sub> も同様に構成される。

#### 【0294】

また、図 21 では、前回の計算部 412 による第 1 の演算の結果得られる各復号途中結果 ( $u_{dv}$ ) が符号ビットを合わせて合計 6 ビット (bit) に量子化され、計算器 415 による第 2 の演算の結果得られる各復号途中結果 ( $v$ ) が 9 ビットに量子化されているものとして、計算器 412<sub>1</sub> を表している。さらに、図 21 の計算器 412<sub>1</sub> には、クロック  $ck$  が供給され、このクロック  $ck$  は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック  $ck$  に同期して処理を行う。

#### 【0295】

図 21 の計算器 412<sub>1</sub> は、制御部 417 から供給される制御信号 D419 に基づいて、復号途中結果格納用メモリ 413 から 1 つずつ読み込まれる、前回の計算部 412 による第 1 の演算の結果得られた復号途中結果 D413 ( $u_{dv}$ ) と、サイクリックシフト回路 411 から 1 つずつ読み込まれる復号途中結果 D411 ( $v$ ) とを用いて、式 (7) と式 (8) にしたがう第 1 の演算を行う。

#### 【0296】

即ち、計算器 412<sub>1</sub> には、サイクリックシフト回路 411 から供給される 5 つの 9 ビットの復号途中結果 D411 ( $v$ ) のうちの、1 つの復号途中結果 D411 が供給されるとともに、復号途中結果格納用メモリ 413 から供給される、前回の計算部 412 による演算の結果である 5 つの 6 ビットの復号途中結果 D413 ( $u_j$ ) のうちの、前回の計算部 412 による演算の結果である 1 つの復号途中結果 D413 が供給され、その 9 ビットの復号途中結果 D411 ( $v$ ) と 6 ビットの復号途中結果 D413 ( $u_{dv}$ ) が、減算器 431 に供給される。また、計算器 412<sub>1</sub> には、制御部 417 から制御信号 D419 が供給され、その制御信号 D419 がセクタ 435 とセクタ 442 に供給される。

#### 【0297】

減算器 431 は、9 ビットの復号途中結果 D411 ( $v$ ) から 6 ビットの復号途中結果 D413 ( $u_j$ ) を減算し、その 6 ビットの減算値 D431 を出力する。即ち、減算器 431 は、式 (8) にしたがって演算を行い、その演算の結果である減算値 D431 ( $v_i$ ) を出力する。

#### 【0298】

減算器 431 により出力された 6 ビットの減算値 D431 のうち、最上位ビットの正負を示す符号ビット D432 ( $\text{sign}(v_i)$ ) が EXOR 回路 440 に供給され、下位 5 ビットの絶対値 D433 ( $|v_i|$ ) が LUT 432 に供給される。

#### 【0299】

LUT 432 は、絶対値 D433 ( $|v_i|$ ) に対して、式 (7) における  $\phi(|v_i|)$  の演算を行った 5 ビットの演算結果 D434 ( $\phi(|v_i|)$ ) を読み出し、加算器 433 と FIFO メモリ 438 に供給する。

#### 【0300】

加算器 433 は、演算結果 D434 ( $\phi(|v_i|)$ ) とレジスタ 434 に格納されている 9 ビットの値 D435 とを加算することにより、演算結果 D434 を積算し、その結果得られる 9 ビットの積算値をレジスタ 434 に再格納する。なお、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D411 から求められた絶対値 D433 ( $|v_i|$ ) に対する演算結果が積算された場合、レジスタ 434 はリセットされる。

#### 【0301】

検査行列の 1 行に亘る復号途中結果 D411 が 1 つずつ読み込まれ、レジスタ 434 に 1 行分の演算結果 D434 が積算された積算値が格納された場合、制御部 417 から供給される制御信号 D419 は、0 から 1 に変化する。例えば、行の重み (row weight) が「9」である場合、制御信号 D419 は、1 から 8 クロック目までは、「0」となり、9 クロック目では「1」となる。

#### 【0302】

制御信号 D419 が「1」の場合、セクタ 435 は、レジスタ 434 に格納されている値、即ち、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D411 (復号途中結果  $v$ ) から求められた  $\phi(|v_i|)$  が積算された 9 ビットの値 D435 ( $i=1$  から  $i=d_c$  までの  $\sum \phi(|v_i|)$ ) を選択し、値 D436 として、レジスタ 436 に出力して格納させる。レジスタ 436 は、格納している値 D436 を、9 ビットの値 D437 として、セクタ 435 と加算器 437 に供給する。制御信号 D419 が「0」の場合、セクタ 435 は、レジスタ 436 から供給された値 D437 を選択し、レジスタ 436 に出力して再格納させる。即ち、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D411 (復号途中結果  $v$ ) から求められた  $\phi(|v_i|)$  が積算されるまで、レジスタ 436 は、前回積算された  $\phi(|v_i|)$  を、セクタ 435 と加算器 437 に供給する。

#### 【0303】

一方、FIFO メモリ 438 は、レジスタ 436 から新たな値 D437 ( $i=1$  から  $i=d_c$  までの  $\sum \phi(|v_i|)$ ) が出力されるまでの間、LUT 432 が出力した演算結果 D434 ( $\phi(|v_i|)$ ) を遅延し、5 ビットの値 D438 として減算器 437 に供給する。減算器 437 は、レジスタ 436 から供給された値 D437 から、FIFO メモリ 438 から供給された値 D438 を減算し、その減算結果を、5 ビットの減算値 D439 として LUT 439 に供給する。即ち、減算器 437 は、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D411 (復号途中結果  $v$ ) から求められた  $\phi(|v_i|)$  の積算値から、求めたい枝に対応する復号途中結果、即ち、検査行列の所定の 1 に対応する復号途中結果 D411 (復号途中結果  $v$ ) から求められた  $\phi(|v_i|)$  を減算して、その減算値 ( $i=1$  から  $i=d_c-1$  までの  $\sum \phi(|v_i|)$ ) を減算値 D439 として LUT 439 に供給する。

#### 【0304】

LUT 439 は、減算値 D439 ( $i=1$  から  $i=d_c-1$  までの  $\sum \phi(|v_i|)$ ) に対して、式 (7) における  $\phi^{-1}(\sum \phi(|v_i|))$  の演算を行った 5 ビットの演算結果 D440 ( $\phi^{-1}(\sum \phi(|v_i|))$ ) を出力する。

#### 【0305】

以上の処理と並行して、EXOR 回路 440 は、レジスタ 441 に格納されている 1 ビットの値 D442 と符号ビット D432 との排他的論理和を演算することにより、符号ビットどうしの乗算を行い、1 ビットの乗算結果 D441 をレジスタ 441 に再格納する。なお、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D411 から求められた符号ビット D432 が乗算された場合、レジスタ 441 はリセットされる。

#### 【0306】

検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D411 から求められた符号ビット D432 が乗算された乗算結果 D441 ( $i=1$  から  $d_c$  までの  $\Pi \text{sign}(v_i)$ ) がレジスタ 441 に格納された場合、制御部 417 から供給される制御信号 D419 は、「0」から「1」に変化する。

#### 【0307】

制御信号 D419 が「1」の場合、セクタ 442 は、レジスタ 441 に格納されている値

、即ち、検査行列の1行に亘る全ての1に対応する復号途中結果D411から求められた符号ビットD432が乗算された値D442 ( $i=1$  から  $i=d_c$  までの  $\Pi \text{sign}(v_i)$ ) を選択し、1ビットの値D443としてレジスタ443に出力して格納させる。レジスタ443は、格納している値D443を、1ビットの値D444としてセレクタ442とEXOR回路445に供給する。制御信号D419が「0」の場合、セレクタ442は、レジスタ443から供給された値D444を選択し、レジスタ443に出力して再格納させる。即ち、検査行列の1行に亘る全ての1に対応する復号途中結果D411 (復号途中結果 $v$ ) から求められた符号ビットD432が乗算されるまで、レジスタ443は、前回格納した値を、セレクタ442とEXOR回路445に供給する。

#### 【0308】

一方、FIFOメモリ444は、レジスタ443から新たな値D444 ( $i=1$  から  $i=d_c$  までの  $\Pi \text{sign}(v_i)$ ) がEXOR回路445に供給されるまでの間、符号ビットD432を遅延し、1ビットの値D445としてEXOR回路445に供給する。EXOR回路445は、レジスタ443から供給された値D444と、FIFOメモリ444から供給された値D445との排他的論理和を演算することにより、値D444を、値D445で除算し、1ビットの除算結果を除算値D446として出力する。即ち、EXOR回路445は、検査行列の1行に亘る全ての1に対応する復号途中結果D411から求められた符号ビットD432 ( $\text{sign}(v_i)$ ) の乗算値を、検査行列の所定の1に対応する復号途中結果D411から求められた符号ビットD432 ( $\text{sign}(v_i)$ ) で除算して、その除算値 ( $i=1$  から  $i=d_c-1$  までの  $\Pi \text{sign}(v_i)$ ) を除算値D446として出力する。

#### 【0309】

計算器412<sub>1</sub>では、LUT439から出力された5ビットの演算結果D440を下位5ビットとするとともに、EXOR回路445から出力された1ビットの除算値D446を最上位ビットとする合計6ビットが復号途中結果D412 (復号途中結果 $u_j$ ) として出力される。

#### 【0310】

以上のように、計算器412<sub>1</sub>では、式(7)と式(8)の演算が行われ、復号途中結果 $u_j$ が求められる。

#### 【0311】

なお、図15の検査行列の行の重みの最大は9であるため、即ち、計算器412<sub>1</sub>に供給される復号途中結果D411 ( $v$ ) と復号途中結果D413 ( $u_{dv}$ ) の最大数は9であるため、計算器412<sub>1</sub>は、9個の復号途中結果D411から求められる9個の演算結果D434 ( $\phi(|v_i|)$ ) を遅延させるFIFOメモリ438と、9個の符号ビットD432を遅延させるFIFOメモリ444を有している。行の重みが9未満の行のメッセージを計算するときには、FIFOメモリ438とFIFOメモリ444における遅延量が、その行の重みの値に減らされる。

#### 【0312】

図22は、図18の計算部415の計算器415<sub>1</sub>の構成例を示すブロック図である。

#### 【0313】

なお、図22では、計算器415<sub>1</sub>について説明するが、計算器415<sub>2</sub>乃至計算器415<sub>5</sub>も同様に構成される。

#### 【0314】

また、図22では、計算器412による第1の演算の結果得られる各復号途中結果 ( $u_j$ ) が符号ビットを合わせて合計6ビットに量子化されているものとして、計算器415<sub>1</sub>を表している。さらに、図22の計算器415<sub>1</sub>には、クロック $ck$ が供給され、このクロック $ck$ は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック $ck$ に同期して処理を行う。

#### 【0315】

図22の計算器415<sub>1</sub>は、制御部417から供給される制御信号D422に基づいて、受信メモリ416から1つずつ読み込まれる受信データD417 (受信値 $u_{0i}$ ) と、サイクリックシフト回路414から1つずつ読み込まれる復号途中結果D414 ( $u_j$ ) とを用いて、式(5)にしたがう第2の演算を行う。

#### 【0316】

即ち、計算器 415<sub>1</sub>では、サイクリックシフト回路 414 から、検査行列の各行の 1 に対応する 6 ビットの復号途中結果 D414 (復号途中結果  $u_j$ ) が 1 つずつ読み込まれ、その復号途中結果 D414 が、加算器 471 に供給される。また、計算器 415<sub>1</sub>では、受信用メモリ 416 から 6 ビットの受信データ D417 が 1 つずつ読み込まれ、加算器 475 に供給される。さらに、計算器 415<sub>1</sub>には、制御部 417 から制御信号 D422 が供給され、その制御信号 D422 は、セクタ 473 に供給される。

**【0317】**

加算器 471 は、復号途中結果 D414 (復号途中結果  $u_j$ ) とレジスタ 472 に格納されている 9 ビットの値 D471 とを加算することにより、復号途中結果 D414 を積算し、その結果得られる 9 ビットの積算値を、レジスタ 472 に再格納する。なお、検査行列の 1 列に亘る全ての 1 に対応する復号途中結果 D414 が積算された場合、レジスタ 472 はリセットされる。

**【0318】**

検査行列の 1 列に亘る復号途中結果 D414 が 1 つずつ読み込まれ、レジスタ 472 に 1 列分の復号途中結果 D414 が積算された値が格納された場合、制御部 417 から供給される制御信号 D422 は、「0」から「1」に変化する。例えば、列の重みが「5」である場合、制御信号 D422 は、1 から 4 クロック目までは「0」となり、5 クロック目では「1」となる。

**【0319】**

制御信号 D422 が「1」の場合、セクタ 473 は、レジスタ 472 に格納されている値、即ち、検査行列の 1 列に亘る全ての枝からの復号途中結果 D414 (復号途中結果  $u_j$ ) が積算された 9 ビットの値 D471 ( $j = 1$  から  $d_v$  までの  $\sum u_j$ ) を選択し、レジスタ 474 に出力して格納させる。レジスタ 474 は、格納している値 D471 を、9 ビットの値 D472 として、セクタ 471 と加算器 475 に供給する。制御信号 D422 が「0」の場合、セクタ 473 は、レジスタ 474 から供給された値 D472 を選択し、レジスタ 474 に出力し再格納させる。即ち、検査行列の 1 列に亘る全ての枝からの復号途中結果 D414 (復号途中結果  $u_j$ ) が積算されるまで、レジスタ 474 は、前回積算された値を、セクタ 473 と加算器 475 に供給する。

**【0320】**

加算器 475 は、9 ビットの値 D472 と、受信用メモリ 416 から供給された 6 ビットの受信データ D417 とを加算して、その結果得られる 6 ビットの値を復号途中結果 D415 (復号途中結果  $v$ ) として出力する。

**【0321】**

以上のように、計算器 415<sub>1</sub>では、式 (5) の演算が行われ、復号途中結果  $v$  が求められる。

**【0322】**

なお、図 8 の検査行列の列の重みの最大は 5 であるため、即ち、計算器 415<sub>1</sub>に供給される復号途中結果  $u_j$  の最大数は 5 であるため、計算器 415<sub>1</sub>は、6 ビットの復号途中結果  $u_j$  を最大 5 個加算する。従って、計算器 415<sub>1</sub>の出力は、9 ビットの値となっている。

**【0323】**

図 23 は、図 18 の復号途中結果格納用メモリ 413 の構成例を示すブロック図である。

**【0324】**

復号途中結果格納用メモリ 413 は、スイッチ 501 と 504、および 2 つのシングルポート RAM である復号途中結果格納用 RAM 502 と 503 から構成される。

**【0325】**

この復号途中結果格納用メモリ 413 の各部について詳細に説明する前に、まず、復号途中結果格納用 RAM 502 と 503 へのデータの格納方法について説明する。

**【0326】**

復号途中結果格納用RAM 502と503は、計算部412による第1の演算の結果得られ、スイッチ501を介して供給された復号途中結果D412を格納する。

**【0327】**

具体的には、復号途中結果格納用RAM 502の第1アドレスから第9アドレスには、図15の検査行列Hの第1行目から第5行目までの1に対応する復号途中結果D412(D501)が、各行ともに横方向(列方向)に詰めた形に(0を無視した形で)格納される。

**【0328】**

即ち、第j行第i列を、(j, i)と表すこととすると、復号途中結果格納用RAM 502の第1アドレスには、図15の検査行列の構成行列である(1,1)から(5,5)の $5 \times 5$ の単位行列の1に対応するデータが、第2アドレスには、図15の検査行列の構成行列である(1,2)から(5,25)のシフト行列( $5 \times 5$ の単位行列を右方向に3つだけサイクリックシフトしたシフト行列)の1に対応するデータが格納される。第3アドレスから第8アドレスも同様に図15の検査行列の構成行列と対応づけてデータが格納される。そして、第9アドレスには、検査行列の(1,86)から(5,90)のシフト行列( $5 \times 5$ の単位行列のうちの1行目の1を0に置き換えて1つだけ左にサイクリックシフトしたシフト行列)の1に対応するデータが格納される。ここで、図15の検査行列の(1,86)から(5,90)のシフト行列においては、1行目に1がないため、第9アドレスにはデータが格納されない。

**【0329】**

復号途中格納用RAM 502の第10アドレスから第18アドレスには、図15の検査行列の第11行目から第15行目までの1に対応するデータが格納される。即ち、第10アドレスには、検査行列の(11,6)から(15,10)の $5 \times 5$ の単位行列を右に3つだけサイクリックシフトした行列の1に対応するデータが格納され、第11アドレスには、検査行列の(11,11)から(15,15)の和行列( $5 \times 5$ の単位行列と、 $5 \times 5$ の単位行列を右に3つだけサイクリックシフトしたシフト行列との和である和行列)を構成するシフト行列の1に対応するデータが格納される。また、第12アドレスには、検査行列の(11,6)から(15,10)の和行列を構成する単位行列の1に対応するデータが格納される。以下、第13アドレスから第18アドレスについても、検査行列に対応づけてデータが格納される。

**【0330】**

即ち、重みが2以上の構成行列については、その構成行列を、重みが1である $P \times P$ の単位行列、そのコンポーネントである1のうち1個以上が0になった準単位行列、または単位行列もしくは準単位行列をサイクリックシフトしたシフト行列のうちの複数の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列の1の位置に対応するデータ(単位行列、準単位行列、またはシフト行列に属する枝に対応するメッセージの復号途中結果)は、同一アドレスに格納される。

**【0331】**

同様に、復号途中格納用RAM 502の第19アドレスから第27アドレスには、図15の検査行列に対応づけて、第21行目から第25行目までの1に対応するデータが格納される。即ち、復号途中結果格納用RAM 502のワード数は、27である。

**【0332】**

復号途中結果格納用RAM 503の第1アドレスから第9アドレスには、図15の検査行列Hの第6行目から第10行目までの1に対応する復号途中結果D412(D502)が、各行ともに横方向(列方向)に詰めた形に(0を無視した形で)格納される。

**【0333】**

即ち、復号途中結果格納用RAM 503の第1アドレスには、検査行列の構成行列である(6,1)から(10,5)の和行列( $5 \times 5$ の単位行列を右に1つだけサイクリックシフトした第1のシフト行列と、右に2つだけサイクリックシフトした第2のシフト行列の和である和行列)を構成する第1のシフト行列の1に対応するデータが、第2アドレスには、検査行列の構成行列である(6,1)から(10,5)の和行列を構成する第2のシフト行列の1に対応するデータが格納される。以下、第3アドレスから第9アドレスも同様に検査行列の構成行列と対応づけてデータが格納される。

## 【0334】

同様に、復号途中格納用RAM 503の第10アドレスから第18アドレスには、図15の検査行列の第16行目から第20行目までの1に対応するデータが、第19アドレスから第27アドレスには、検査行列の第26行目から第30行目までの1に対応するデータが、図15の検査行列に対応づけて格納される。即ち、復号途中結果格納用RAM 503のワード数は、27である。

## 【0335】

上述したように、復号途中結果格納用RAM 502と503のワード(word)数は、27である。即ち、ワード数は、検査行列の行の重み(row weight)の9と行数の30とを乗算し、その乗算結果(検査行列の1の数)を、同時に読み出す復号途中結果D501の数の5で除算し、さらに、復号途中結果格納用メモリ413が有する復号途中結果格納用RAMの個数の2で除算した値となる。

## 【0336】

以下、図23の復号途中結果格納用メモリ413の各部の動作について詳細に説明する。

## 【0337】

復号途中結果格納用メモリ413には、計算部412により第1の演算が行われる場合、計算部412から第1の演算の結果得られる復号途中結果D412 ( $u_j$ ) が供給され、その復号途中結果D412が復号途中結果格納用RAM 502または復号途中結果格納用RAM 503のうちの一方の所定のアドレスに書き込まれると同時に、他方から、前回の計算部412による第1の演算の結果得られた復号途中結果D412 ( $u_j$ ) が読み出され、計算部412に出力される。一方、計算部415により第2の演算が行われる場合、復号途中結果格納用メモリ413は、復号途中結果格納用RAM 502または復号途中結果格納用RAM 503に書き込みを行わず、どちらか一方のRAMの所定のアドレスから復号途中結果を読み出して、サイクリックシフト回路414に供給する。

## 【0338】

スイッチ501には、計算部412から5つの復号途中結果D412が供給されるとともに、その復号途中結果D412を書き込むメモリとして、復号途中結果格納用RAM 502または復号途中結果格納用RAM 503の一方の選択を表す制御信号D420<sub>1</sub>が制御部417から供給される。スイッチ501は、制御信号D420<sub>1</sub>に基づいて、復号途中結果格納用RAM 502または復号途中結果格納用RAM 503の一方を選択し、その選択した一方に、5つの復号途中結果D412を供給する。

## 【0339】

復号途中結果格納用RAM 502には、スイッチ501から5つの復号途中結果D412が、復号途中結果D501として供給されるとともに、制御部417からアドレスを表す制御信号D420<sub>2</sub>が供給される。復号途中結果格納用RAM 502は、制御信号D420<sub>2</sub>が表すアドレスに既に格納されている前回の計算部412による第1の演算の結果得られた5つの復号途中結果D501を読み出し、復号途中結果D503としてスイッチ504に供給する。また、復号途中結果格納用RAM 502は、制御信号D420<sub>2</sub>が表すアドレスに、スイッチ501から供給された5つの復号途中結果D501を格納する(書き込む)。

## 【0340】

復号途中結果格納用RAM 503には、スイッチ501から5つの復号途中結果D412が、復号途中結果D502として供給されるとともに、制御部417からアドレスを表す制御信号D420<sub>3</sub>が供給される。復号途中結果格納用RAM 503は、制御信号D420<sub>3</sub>が表すアドレスに既に格納されている前回の計算部412による第1の演算の結果得られた5つの復号途中結果D502を読み出し、復号途中結果D504としてスイッチ504に供給する。また、復号途中結果格納用RAM 502は、制御信号D420<sub>3</sub>が表すアドレスに、スイッチ501から供給された5つの復号途中結果D502を格納する(書き込む)。

## 【0341】

スイッチ504には、復号途中結果格納用RAM 502から復号途中結果D503が供給され

るか、あるいは復号途中結果格納用RAM503から復号途中結果D504が供給される。また、制御部417から、復号途中結果格納用RAM502または復号途中結果格納用RAM503の一方の選択を表す制御信号D420<sub>4</sub>が供給される。スイッチ504は、制御信号D420<sub>1</sub>に基づいて、復号途中結果格納用RAM502または復号途中結果格納用RAM503の一方を選択し、その選択した一方から供給された5つの復号途中結果を、5つの復号途中結果D413として計算部412とサイクリックシフト回路414に供給する。

#### 【0342】

図24は、復号途中結果格納用メモリ413の復号途中結果格納用RAM502と復号途中結果格納用RAM503の読み出しと書き込みの動作を説明するタイミングチャートである。

#### 【0343】

なお、図24において、横軸は、時間(t)を表している。

#### 【0344】

復号途中結果格納用メモリ413では、計算部412により第1の演算が行われる場合、復号途中結果格納用RAM502が、制御信号D420<sub>2</sub>に基づいて、既に格納している、前回の計算部412の第1の演算の結果得られた復号途中結果D501のうち、同一アドレスに格納している検査行列の第1行目から第5行目までの1に対応する復号途中結果D501を、5つ単位で9回読み出し、スイッチ504を介して、計算部412に供給する。即ち、図15の検査行列Hの行重みは、9であるため、検査行列Hの各行の1に対応する復号途中結果は9つあり、復号途中結果格納用RAM502は、第1行目から第5行目までの1に対応する5つの復号途中結果D501を、5つ単位で9回読み出す。

#### 【0345】

次に、復号途中結果格納用RAM503は、制御信号D420<sub>3</sub>に基づいて、既に格納している、前回の計算部412による第1の演算の結果得られた復号途中結果D502のうち、同一アドレスに格納している検査行列の第6行目から第10行目までの1に対応する復号途中結果D502を、5つ単位で9回続けて読み出し、スイッチ504を介して、計算部412に供給する。それと同時に、復号途中結果格納用RAM502には、計算部412により、いま行われている第1の演算の結果得られる検査行列の第1行目から第5行目までの1に対応する5つの復号途中結果D412がスイッチ501を介して、復号途中結果D501として供給され、復号途中結果格納用RAM502は、その復号途中結果D501を、制御信号D420<sub>2</sub>に基づいて、既に読み出された復号途中結果D503が格納されていたアドレスに9回続けて格納する。

#### 【0346】

その後、復号途中結果格納用RAM502は、制御信号D420<sub>2</sub>に基づいて、既に格納している、前回の計算部412による第1の演算の結果得られた復号途中結果D501のうち、同一アドレスに格納している検査行列の第11行目から第15行目までの1に対応する復号途中結果D501を、5つ単位で9回続けて読み出し、スイッチ504を介して、計算部412に供給する。それと同時に、復号途中結果格納用RAM503には、計算部412により、いま行われている第1の演算の結果得られる検査行列の第6行目から第10行目までの1に対応する5つの復号途中結果D412がスイッチ501を介して、復号途中結果D502として供給され、復号途中結果格納用RAM503は、その復号途中結果D502を、制御信号D420<sub>3</sub>に基づいて、既に読み出された復号途中結果D504が格納されていたアドレスに9回続けて格納する。

#### 【0347】

以後、同様に、計算部412による第1の演算の結果得られる検査行列の全ての1に対応する復号途中結果が、復号途中結果格納用RAM502または復号途中結果格納用RAM503に格納されるまで、復号途中結果格納用RAM502と復号途中結果格納用RAM503は、9回ずつの読み出しまたは書き込みを交互に行う。

#### 【0348】

復号途中結果格納用メモリ413では、計算部415による第2の演算が行われる場合

、制御信号D420<sub>2</sub>に基づいて、復号途中結果格納用RAM 502から既に格納されている第1の演算の結果得られる復号途中結果D503を読み出すか、あるいは制御信号D420<sub>3</sub>に基づいて、復号途中結果格納用RAM 503から、既に格納されている第1の演算の結果得られる復号途中結果D504を読み出し、その読み出した復号途中結果をスイッチ504を介して、サイクリックシフト回路414に供給する。

**【0349】**

図25は、図18の復号装置400の復号処理を説明するフローチャートである。この処理は、例えば、受信用メモリ416に復号すべき受信データが格納されたとき、開始される。

**【0350】**

ステップS50において、サイクリックシフト回路414は、復号途中結果格納用メモリ413から供給された後述するステップS56で格納される5つの復号途中結果D413を、サイクリックシフトし、計算部415に供給する。

**【0351】**

具体的には、サイクリックシフト回路414には、復号途中結果格納用メモリ413から5つの復号途中結果D413が供給されるとともに、制御部417から、その復号途中結果D413に対応する検査行列の1が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報（Matrixデータ）を表す制御信号D421が供給される。サイクリックシフト回路414は、制御信号D421を元に、5つの復号途中結果D413をサイクリックシフトし（並べ替え）、その結果を復号途中結果D414として、計算部415に供給する。

**【0352】**

なお、受信用メモリ416から供給された受信データD417に対して、まだ第1の演算が行われておらず、復号途中結果格納用メモリ413に復号途中結果D413が格納されていない場合、計算部415は、復号途中結果 $u_j$ を初期値に設定する。

**【0353】**

ステップS51において、計算部415は、第2の演算を行い、その演算の結果である復号途中結果D415を復号途中結果格納用メモリ410に供給する。

**【0354】**

具体的には、計算部415には、ステップS50でサイクリックシフト回路414から5つの復号途中結果D414が供給されるとともに、受信データ用メモリ416から5つの受信データD417が供給され、復号途中結果D415と受信データD417が、計算部415の計算器415<sub>1</sub>乃至415<sub>5</sub>それぞれに1つずつ供給される。さらに、計算部415には、制御部417から制御信号D422が供給され、その制御信号D422が計算器415<sub>1</sub>乃至415<sub>5</sub>に供給される。

**【0355】**

計算器415<sub>1</sub>乃至415<sub>5</sub>は、復号途中結果D414と受信データD417を用いて、制御信号D422に基づいて、式（5）にしたがって、それぞれ演算を行い、その演算の結果得られる検査行列の列に対応する復号途中結果D415(v)を復号途中結果格納用メモリ410に供給する。

**【0356】**

ステップS51の処理後は、ステップS52に進み、復号途中結果格納用メモリ410は、ステップS51で計算部415から供給された復号途中結果D415を、同一アドレスに格納し、ステップS53に進む。

**【0357】**

ステップS53において、制御部417は、計算部415により、検査行列の列に対応する全ての復号途中結果D415が演算されたかどうかを判定し、全ての復号途中結果D415が演算されていないと判定した場合、ステップS50に戻り、上述した処理を繰り返す。

**【0358】**

一方、ステップS53において、制御部417は、計算部415により、検査行列の列



に対応する全ての復号途中結果D415が演算されたと判定した場合、ステップS54に進み、サイクリックシフト回路411は、復号途中結果格納用メモリ410から供給される復号途中結果D410(v)をサイクリックシフトする。

#### 【0359】

具体的には、サイクリックシフト回路411には、復号途中結果格納用メモリ410から5つの復号途中結果D410が供給されるとともに、制御部417から、その復号途中結果D410に対応する検査行列の1が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報(Matrixデータ)を表す制御信号D418が供給される。サイクリックシフト回路411は、制御信号D418を元に、5つの復号途中結果D410をサイクリックシフトし(並べ替え)、その結果を復号途中結果D411として、計算部412に供給する。

#### 【0360】

ステップS54の処理後は、ステップS55に進み、計算部412は、第1の演算を行い、その演算結果である復号途中結果D412をサイクリックシフト回路414に供給する。

#### 【0361】

具体的には、計算部412には、ステップS54でサイクリックシフト回路411から5つの復号途中結果D411(v)が供給されるとともに、後述するステップS56で既に格納された前回の計算部412による第1の演算の結果得られた5つの復号途中結果D412(D413)( $u_j$ )が供給され、その復号途中結果D411と復号途中結果D413が、計算部412の計算器412<sub>1</sub>乃至412<sub>5</sub>のそれぞれに1つずつ供給される。さらに、計算部412には、制御部417から制御信号D419が供給され、その制御信号D419が計算器412<sub>1</sub>乃至412<sub>5</sub>に供給される。

#### 【0362】

計算器412<sub>1</sub>乃至412<sub>5</sub>は、それぞれ復号途中結果D411と復号途中結果D413とを用いて、制御信号D419に基づいて、式(7)と式(8)にしたがって、それぞれ演算を行い、その演算の結果得られる復号途中結果D412( $u_j$ )を復号途中結果格納用メモリ413に供給する。

#### 【0363】

ステップS55の処理後は、ステップS56に進み、復号途中結果格納用メモリ413は、ステップS55で計算部412から供給された5つの復号途中結果D412を、同一のアドレスに格納し、ステップS57に進む。

#### 【0364】

ステップS57において、制御部417は、計算部412により、検査行列の全ての1に対応する復号途中結果D412が演算されたかどうかを判定し、全ての復号途中結果が演算されていないと判定した場合、ステップS54に戻り、上述した処理を繰り返す。

#### 【0365】

一方、ステップS57において、制御部417は、計算部412により、全ての1に対応する復号途中結果D412が演算されたと判定した場合、処理を終了する。

#### 【0366】

なお、復号装置400は、復号回数だけ図25の復号処理を繰り返し行ない、最後の第2の演算の結果得られるメッセージD415が、最終的な復号結果として出力される。

#### 【0367】

上述した説明では、復号途中結果格納用メモリ413は、2つのシングルポートRAMから構成にしたが、1つのRAMに対して読み出しと書き込みが同時に起こらないようにすれば、3つ以上のRAMから構成してもよい。例えば、RAMの物理的なビットが足りない場合には、複数のRAMを用いて同じ制御信号を与えることで、論理的に1つのRAMとみなすことができる。

#### 【0368】

また、枝データ(枝に対応するメッセージ)が欠けている箇所に関しては、メモリ格納時(復号途中結果格納用メモリ410と413へのデータ格納時)には、何のメッセージ

も格納せず、また、演算時（計算部 412 での第 1 の演算時と計算部 415 での第 2 の演算時）にも何の演算も行わない。

**【0369】**

図 26 は、本発明を適用した図 15 の検査行列で表現される LDPC 符号を復号する復号装置の他の一実施の形態の構成例を示すブロック図である。

**【0370】**

図 26 の復号装置 600 では、図 16 の枝データ格納メモリ 316 が、枝データ格納メモリ 316 に比べて容量の小さい復号途中結果格納用メモリ 613 になっている。

**【0371】**

復号装置 600 は、復号途中結果格納用メモリ 610、サイクリックシフト回路 611、5 つの計算器 612<sub>1</sub> 乃至計算器 612<sub>5</sub> からなる計算部 612、復号途中結果格納用メモリ 613、サイクリックシフト回路 614、5 つの計算器 615<sub>1</sub> 乃至計算器 615<sub>5</sub> からなる計算部 615、受信用メモリ 616、および制御部 617 から構成される。

**【0372】**

ここで、図 27 乃至図 30 を用いて、図 26 の計算部 612 の計算器 612<sub>1</sub> 乃至計算器 612<sub>5</sub>、および図 30 の計算部 615 の計算器 615<sub>1</sub> 乃至計算器 615<sub>5</sub> と、図 10 のチェックノード計算器 101 および図 11 のバリエブルノード計算器 103 との関係について説明する。

**【0373】**

図 27 と図 28 は、前述の図 10 のチェックノード計算器 101 と図 11 のバリエブルノード計算器 103 とそれぞれ同一の図である。また、図 29 は、計算器 612<sub>k</sub> ( $k=1, 2, \dots, 5$ ) の構成例を示しており、図 30 は、計算器 615<sub>k</sub> ( $k=1, 2, \dots, 5$ ) の構成例を示している。

**【0374】**

図 26 の復号装置 600 では、計算器 612<sub>k</sub> がチェックノード演算を行い、計算部 615<sub>k</sub> が、バリエブルノード演算をおこなうのではなく、計算器 612<sub>k</sub> がチェックノード演算の一部を、計算器 615<sub>k</sub> がチェックノード演算の他の一部とバリエブルノード演算を行う。

**【0375】**

即ち、図 29 の計算器 612<sub>k</sub> は、ブロック D' と E' から構成される。ブロック D' は、図 27 のチェックノード計算器 101 の一部である、検査行列の各列の全ての枝に対応するメッセージ  $v_i$  の絶対値に対して  $\phi$  の演算を行った値を積算するブロック D と同様に構成されている。また、ブロック E' は、検査行列の各列の全ての枝に対応するメッセージ  $v_i$  の符号ビットを乗算するブロック E と同様に構成されている。

**【0376】**

一方、図 30 の計算器 615<sub>k</sub> は、ブロック F'、G'、H' とから構成される。ブロック F' は、図 19 のチェックノード計算器 101 の他の一部である、検査行列の各列の全ての枝に対応するメッセージ  $v_i$  の符号ビットの乗算値から、求めたい枝に対応するメッセージ  $v_i$  の符号ビットを除算するとともに、検査行列の各列の全ての枝に対応するメッセージ  $v_i$  の絶対値に対して  $\phi$  の演算を行った値の積算値から、求めたい枝に対応するメッセージ  $v_i$  の絶対値に対して  $\phi$  の演算を行った値を減算した値に対して、 $\phi^{-1}$  の演算を行うブロック F と同様に構成されている。また、ブロック G' は、メッセージ  $v_i$  の絶対値に対して  $\phi$  の演算を行うブロック G と同様に構成され、ブロック H' は、図 20 のバリエブルノード計算器 103 のバリエブルノード演算を行うブロック H と同様に構成されている。

**【0377】**

そして、図 29 の計算器 612<sub>k</sub> は、ブロック A とブロック B による演算の結果、即ち、チェックノード演算の一部を行った復号途中結果  $w$  を復号途中結果格納用メモリ 613 に供給し、図 30 の計算器 615<sub>k</sub> は、チェックノード演算の他の一部とバリエブルノード演算を行った復号途中結果  $v_i'$  を復号途中結果格納用メモリ 610 に供給する。

**【0378】**

従って、図26の復号装置600は、計算器612<sub>k</sub>の演算と計算器615<sub>k</sub>の演算とを交互に行うことにより、チェックノード演算とバリアブルノード演算を行い、復号を行うことができる。

#### 【0379】

なお、図30の計算器615<sub>k</sub>では、復号途中結果格納用メモリ610に格納されている求めたい枝に対応する復号途中結果 $v_i'$ を用いて、ブロックCで、計算器612<sub>k</sub>の演算の結果得られる復号途中結果 $w$ の絶対値から、求めたい枝に対応する復号途中結果 $v_i'$ を減算するとともに、復号途中結果 $w$ の符号ビットと、求めたい枝に対応する復号途中結果 $v_i'$ の符号ビットを乗算するので、図27のFIFOメモリ127とFIFOメモリ133が必要ない。

#### 【0380】

次に、計算部612の計算器612<sub>1</sub>乃至計算器612<sub>5</sub>で行われる演算と、計算部615の計算器615<sub>1</sub>乃至計算器615<sub>5</sub>で行われる演算について、式を用いて説明する。

#### 【0381】

計算部612は、以下の式(9)にしたがう第1の演算を行い、その第1の演算の結果である復号途中結果 $w$ を復号途中結果格納用メモリ613に供給して格納させる。計算部615は、上述した式(1)と、以下の式(10)と(11)にしたがう第2の演算を行い、その第2の演算の結果である復号途中結果 $v_i'$ を復号途中結果格納用メモリ610に供給して格納させる。

#### 【0382】

##### 【数9】

$$w = \sum_{i=1}^{d_c} |v_i'| \times \prod_{i=1}^{d_c} \text{sign}(v_i') \quad \dots (9)$$

#### 【0383】

##### 【数10】

$$u_j = \phi^{-1}(|w| - |v_i'|) \times \text{sign}(v_i') \times \text{sign}(w) \quad \dots (10)$$

#### 【0384】

##### 【数11】

$$v_i' = \phi(|v_i|) \times \text{sign}(v_i) \quad \dots (11)$$

#### 【0385】

即ち、式(9)にしたがう第1の演算の結果得られる復号途中結果 $w$ は、式(1)、式(10)、式(11)にしたがう第2の演算の結果得られる検査行列 $H$ の $j$ 行のすべての1に対応するチェックノード演算の復号途中結果 $v_i'$ の絶対値 $|v_i'|$ の総和と符号ビット $\text{sign}(v_i')$ の乗算値とを乗算したものであるので、上述した式(7)にしたがうチェックノード演算によって求められる $u_j$ は、式(10)に示すように、式(9)にしたがう第1の演算の結果得られる復号途中結果 $w$ の絶対値 $|w|$ から、検査行列 $H$ の $j$ 行の、各列の“1”(枝)に対応する(複数の)復号途中結果 $v_i'$ のうち、メッセージを求めたい枝に対応する復号途中結果 $v_i'$ の絶対値 $|v_i'|$ を引いた値を用いて表すことができる。

#### 【0386】

復号装置600では、計算部612による式(9)にしたがう第1の演算と、計算部615による式(1)、式(10)、式(11)にしたがう第2の演算とが交互に行われ、計算部615が、最後の第1の演算の結果を用いて、式(5)にしたがう演算を行い、その演算結果を復号結果として出力することにより、LDPC符号の繰返し復号を行う。

#### 【0387】

即ち、復号装置 600 では、計算部 612 が、計算部 615 による第 2 の演算の結果である検査行列  $H$  の  $j$  行のすべての 1 に対応する復号途中結果  $v_i'$  を用いて、第 1 の演算を行い、その演算の結果得られる検査行列の各行に対応する復号途中結果  $w$  を復号途中結果格納用メモリ 613 に格納する。従って、復号途中結果格納用メモリ 613 の容量は、検査行列の “1” の数より少ない検査行列の行数と復号途中結果  $w$  の量子化ビット数とを乗算した値となる。なお、計算部 615 は、計算部 612 による第 1 の演算の結果である検査行列  $H$  の  $i$  列の各行に対応する復号途中結果  $w$  と受信値  $u_{0i}$  を用いて、第 2 の演算を行い、その演算の結果得られる検査行列の  $i$  列の 1 (枝) に対応するチェックノード演算の復号途中結果  $v_i'$  を復号途中結果格納用メモリ 610 に格納する。従って、復号途中結果格納用メモリ 610 に必要な容量は、バリアブルノード演算の結果を格納する図 16 の枝データ格納メモリ 311 と同様に、検査行列の 1 の数と復号途中結果  $v_i'$  の量子化ビット数とを乗算した値となる。

#### 【0388】

従って、復号装置 600 では、図 16 の枝データ格納メモリ 311 に比べて、復号途中結果格納用メモリ 610 のメモリの容量を削減することができ、これにより、復号装置 600 の回路規模を小さくすることができる。

#### 【0389】

以下、図 26 の復号装置 600 の各部の動作について詳細に説明する。

#### 【0390】

復号途中結果格納用メモリ 610 は、制御信号 D618 に基づいて、計算部 615 から供給される 5 つの復号途中結果 D615 をまとめて格納すると同時に、既に格納してある 5 つの復号途中結果 D615 を読み出し、復号途中結果 D610 として、サイクリックシフト回路 611 と計算部 615 に供給する。即ち、復号途中結果格納用メモリ 610 は、サイクリックシフト回路 611 に供給する復号途中結果 D610 の読み出しと、計算部 615 から供給される復号途中結果 D615 の書き込みとを、同時に行う。

#### 【0391】

なお、復号途中結果格納用メモリ 610 には、計算部 615 の第 2 の演算により演算された検査行列の 1 (枝) に対応する復号途中結果  $v_i'$  (第 2 の復号途中結果) が格納されるので、復号途中結果格納用メモリ 610 に格納されるデータ量、即ち、復号途中結果格納用メモリ 610 に必要とされる記憶容量は、復号途中結果の量子化ビット数と、検査行列の 1 の数 (全枝数) との乗算値となる。

#### 【0392】

復号途中結果格納用メモリ 610 は、例えば、5 つの復号途中結果を同時に読み書き可能な、2 つのシングルポート RAM から構成される。復号途中結果格納用メモリ 610 には、計算部 615 から 5 つの復号途中結果 D615 が供給されるとともに、制御部 617 から復号途中結果 D615 の読み書きを制御する制御信号 D618 が供給される。

#### 【0393】

サイクリックシフト回路 611 には、復号途中結果格納用メモリ 610 から 5 つの復号途中結果 D610 が供給されるとともに、制御部 617 から、その復号途中結果 D610 に対応する検査行列の 1 が、検査行列において元となる単位行列などを幾つサイクリックシフトであるかの情報 (Matrix データ) を表す制御信号 D619 が供給される。サイクリックシフト回路 611 は、制御信号 D619 を元に、5 つの復号結果 D610 を並べ替えるサイクリックシフトを行い、その結果を復号途中結果 D611 として、計算部 612 に供給する。

#### 【0394】

計算部 612 は、5 つの計算器 612<sub>1</sub> 乃至 612<sub>5</sub> からなる。計算部 612 には、サイクリックシフト回路 611 から 5 つの復号途中結果 D611 (第 2 の復号途中結果) ( $v_i'$ ) が供給され、その 5 つの復号途中結果 D611 (第 1 の復号途中結果) ( $w$ ) が、計算器 612<sub>1</sub> 乃至 612<sub>5</sub> のそれぞれに供給される。また、計算部 612 には、制御部 617 から制御信号 D620 が供給され、その制御信号 D620 が、計算器 612<sub>1</sub> 乃至 612<sub>5</sub> に供給される。なお、制御信号 D620 は、5 つの計算器 612<sub>1</sub> 乃至 612<sub>5</sub> に共通の信号である。

## 【0395】

計算器612<sub>1</sub>乃至612<sub>5</sub>は、それぞれ復号途中結果D611を用いて、式(9)にしたがって第1の演算を行い、復号途中結果D612(w)を求める。計算部612は、計算器612<sub>1</sub>乃至612<sub>5</sub>による演算の結果得られる5つの復号途中結果D612を復号途中結果格納用メモリ613に供給する。

## 【0396】

復号途中結果格納用メモリ613には、計算部612から、計算部612による第1の演算の結果である検査行列の行に対応する5つの復号途中結果D612が供給され、復号途中結果格納用メモリ613は、計算部612から供給された5つの復号途中結果D612を、第1アドレスから順に格納(記憶)する。

## 【0397】

即ち、復号途中結果格納用メモリ613の第1アドレスには、検査行列の行に対応する復号途中結果のうち、第1行目から第5行目の復号途中結果wが格納される。そして、同様に、第2アドレスには、第6行目から第10行目の復号途中結果wが格納され、第3アドレスには、第11行目から第15行目の復号途中結果wが格納される。以後、同様に、第16行目から第30行目までの復号途中結果wが、5個ずつ、第4アドレスから第6アドレスまで格納され、計60個の復号途中結果wが復号途中結果格納用メモリ613に格納される。従って、復号途中結果格納用メモリ610のワード(word)数は、図15の検査行列Hの行数である30を、同時に読み書きする復号途中結果の数である5で割り算した6となる。

## 【0398】

また、復号途中結果格納用メモリ613は、既に格納してある5つの復号途中結果D613から、計算部615が求めようとする復号途中結果v<sub>i</sub>'の対応する検査行列Hの列において“1”になっている復号途中結果wを5つ同時に読み出し、復号途中結果D613として、サイクリックシフト回路614に供給する。

## 【0399】

なお、復号途中結果格納用メモリ613は、例えば、5つの復号途中結果を同時に読み書き可能な、シングルポートRAMで構成される。また、復号途中結果格納用メモリ613には、計算部612の第1の演算により演算された行に対応する復号途中結果wが格納されるので、復号途中結果格納用メモリ613に格納されるデータ量、即ち、復号途中結果格納用メモリ613に必要とされる記憶容量は、復号途中結果の量子化ビット数と、検査行列Hの行数との乗算値である。

## 【0400】

サイクリックシフト回路614には、復号途中結果格納用メモリ613から5つの復号途中結果D613(復号途中結果w)が供給されるとともに、制御部617から、その復号途中結果D613に対応する検査行列の1が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報(Matrixデータ)を表す制御信号D621が供給される。サイクリックシフト回路614は、制御信号D621を元に、5つの復号途中結果D613を並べ替えるサイクリックシフトを行い、その結果を復号途中結果D614として、計算部615に供給する。

## 【0401】

計算部615は、5つの計算器615<sub>1</sub>乃至615<sub>5</sub>からなる。バリエブルノード計算部615には、サイクリックシフト回路614から5つの復号途中結果D614(w)が供給されるとともに、復号途中結果格納用メモリ610から5つの復号途中結果D610(v<sub>i</sub>')が供給され、その復号途中結果D614と復号途中結果D610が、計算器615<sub>1</sub>乃至615<sub>5</sub>のそれぞれに供給される。また、計算部615には、受信用メモリ617から5つの受信データD617が供給され、その受信データD617が、計算器615<sub>1</sub>乃至615<sub>5</sub>のそれぞれに供給される。さらに、計算部617には、制御部617から制御信号D622が供給され、その制御信号D622が計算器615<sub>1</sub>乃至615<sub>5</sub>に供給される。なお、制御信号D622は、5つの計算器617<sub>1</sub>乃至617<sub>5</sub>に共通の信号である。

## 【0402】

計算器 615<sub>1</sub>乃至 615<sub>5</sub>は、それぞれ復号途中結果D614とD611、受信データD617 (LD PC符号) とを用いて、式(1)、式(10)、式(11)にしたがって、それぞれ第2の演算を行い、検査行列の各列の1に対応する5つの復号途中結果D615 ( $v_i'$ ) を求める。計算部 615は、計算器 615<sub>1</sub>乃至 615<sub>5</sub>の第2の演算の結果得られる5つの復号途中結果D615を、復号途中結果格納用メモリ 610に供給する。

## 【0403】

受信用メモリ 616は、通信路を通して受信した受信値 (符号ビット) D616から計算した符号ビットの0らしさの値である受信LLR (対数尤度比) を、受信データD617として格納する。

## 【0404】

即ち、受信用メモリ 616の第1のアドレスには、検査行列の列に対応する受信データD617のうち、検査行列の第1列目から第5列目までに対応する受信データD617が格納される。そして、第2のアドレスには、検査行列の第6列目から第10列目までに対応する受信データD617が格納され、第3アドレスには、検査行列の第11列目から第16列目までに対応する受信データD617が格納される。以後、同様に、第4アドレスから第18アドレスまでに、検査行列の第17列目から第90列目までに対応する受信データD617が、5つずつ格納される。

## 【0405】

そして、受信用メモリ 616は、既に格納している受信データD617を計算部 615による第2の演算に必要な順番に5つずつ同時に読み出し、計算部 615に供給する。

## 【0406】

なお、受信用メモリ 616は、例えば、シングルポートRAMから構成される。また、受信用メモリ 616に格納されるデータ量、即ち、受信用メモリ 616に必要なとされる記憶容量は、LDPC符号の符号長と、受信データの量子化ビット数との乗算値である。さらに、受信用メモリ 616のワード (word) 数は、LDPC符号の符号長、即ち、検査行列の列数である90を、同時に読み出す受信データD617の数である5で割り算した値の18である。

## 【0407】

制御部 617は、制御信号D618を復号途中結果格納用メモリ 610に、制御信号D619をサイクリックシフト回路 611に供給することにより、それぞれを制御する。また、制御部 617は、制御信号D620を計算部 612に、制御信号D621をサイクリックシフト回路 614に、制御信号D622を計算部 615に供給することにより、それぞれを制御する。

## 【0408】

復号途中結果格納用メモリ 610、サイクリックシフト回路 611、計算部 612、復号途中結果格納用メモリ 613、サイクリックシフト回路 614、計算部 615の順で、データが一巡することで、復号装置 600は、1回の復号を行うことができる。復号装置 600では、所定の回数だけ繰り返して復号が行われた後、計算部 615が、式(5)にしたがう演算を行い、その演算結果が最終的な復号結果として出力される。

## 【0409】

図29は、図26の計算部 612の計算器 612<sub>1</sub>の構成例を示すブロック図である。

## 【0410】

なお、図29では、計算器 612<sub>1</sub>について説明するが、計算器 612<sub>2</sub>乃至計算器 612<sub>5</sub>も同様に構成される。

## 【0411】

また、図29では、計算器 615による第2の演算の結果得られる各復号途中結果 ( $v_i'$ ) が6ビットに量子化されているものとして、計算器 612<sub>1</sub>を表している。さらに、図29の計算器 612<sub>1</sub>には、クロック  $ck$  が供給され、このクロック  $ck$  は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック  $ck$  に同期して処理を行う。

## 【0412】

図29の計算器612<sub>1</sub>は、制御部617から供給される制御信号D620に基づいて、サイクリックシフト回路611から1つずつ読み込まれる復号途中結果D611 ( $v_i'$ ) を用いて、式(9)にしたがう第1の演算を行う。

**【0413】**

即ち、計算器612<sub>1</sub>には、サイクリックシフト回路611から供給される5つの6ビットの復号途中結果D611( $v_i'$ )のうちの、1つの復号途中結果D611が供給され、最上位ビットの符号ビットD631がEXOR回路635に供給されるとともに、その6ビットの復号途中結果D611 ( $v_i'$ ) の下位5ビットの絶対値D632 ( $|v_i'|$ ) が、加算器631に供給される。また、計算器612<sub>1</sub>には、制御部617から制御信号D620が供給され、その制御信号D620がセクタ633とセクタ637に供給される。

**【0414】**

加算器631は、絶対値D632 ( $|v_i'|$ ) とレジスタ632に格納されている9ビットの値D633とを加算することにより、絶対値D632 ( $|v_i'|$ ) を積算し、その結果得られる9ビットの積算値をレジスタ632に再格納する。なお、検査行列の1行に亘る全ての1に対応する復号途中結果D611から求められた絶対値D632 ( $|v_i'|$ ) が積算された場合、レジスタ632はリセットされる。

**【0415】**

検査行列の1行に亘る復号途中結果D611が1つずつ読み込まれ、レジスタ632に1行分の絶対値D632が積算された積算値が格納された場合、制御部617から供給される制御信号D620は、0から1に変化する。例えば、行の重み (row weight) が「9」である場合、制御信号D620は、1から8クロック目までは、「0」となり、9クロック目では「1」となる。

**【0416】**

制御信号D620が「1」の場合、セクタ633は、レジスタ632に格納されている値、即ち、検査行列の1行に亘る全ての1に対応する復号途中結果D611 (復号途中結果 $v_i'$ ) の絶対値D632 ( $|v_i'|$ ) が積算された9ビットの値D633 ( $i=1$  から  $i=d_c$  までの  $\sum |v_i'|$ ) を選択し、値D634として、レジスタ634に出力して格納させる。レジスタ634は、格納している値D634を、9ビットの値D635として、セクタ633に供給するとともに、出力する。制御信号D620が「0」の場合、セクタ633は、レジスタ634から供給された値D635を選択し、レジスタ634に出力して再格納させる。即ち、検査行列の1行に亘る全ての1に対応する復号途中結果D611 (復号途中結果 $v_i'$ ) の絶対値D632 ( $|v_i'|$ ) が積算されるまで、レジスタ634は、前回積算された $|v_i'|$ を、セクタ633に供給するとともに、出力する。

**【0417】**

以上の処理と並行して、EXOR回路635は、レジスタ636に格納されている1ビットの値D637と符号ビットD631との排他的論理和を演算することにより、符号ビットどうしの乗算を行い、1ビットの乗算結果D636をレジスタ636に再格納する。なお、検査行列の1行に亘る全ての1に対応する復号途中結果D611の符号ビットD631が乗算された場合、レジスタ636はリセットされる。

**【0418】**

検査行列の1行に亘る全ての1に対応する復号途中結果D611から求められた符号ビットD631が乗算された乗算結果D636 ( $i=1$  から  $d_c$  までの  $\prod \text{sign}(v_i')$ ) がレジスタ636に格納された場合、制御部617から供給される制御信号D620は、「0」から「1」に変化する。

**【0419】**

制御信号D620が「1」の場合、セクタ637は、レジスタ636に格納されている値、即ち、検査行列の1行に亘る全ての1に対応する復号途中結果D611の符号ビットD631が乗算された値D637 ( $i=1$  から  $i=d_c$  までの  $\prod \text{sign}(v_i')$ ) を選択し、1ビットの値D638としてレジスタ638に出力して格納させる。レジスタ638は、格納している値D638と、1ビットの値D639としてセクタ637に供給するとともに、出力する。制御信号D620

が「0」の場合、セクタ637は、レジスタ638から供給された値D639を選択し、レジスタ638に出力して再格納させる。即ち、検査行列の1行に亘る全ての1に対応する復号途中結果D611（復号途中結果 $v_i'$ ）の符号ビットD631が乗算されるまで、レジスタ638は、前回格納した値を、セクタ637に供給するとともに、出力する。

#### 【0420】

計算器612<sub>i</sub>では、レジスタ634から出力された9ビットの値D635（ $i=1$ から $i=d_c$ までの $\sum |v_i'|$ ）を下位9ビットとするとともに、レジスタ638から出力された1ビットの値D639（ $\text{sign}(v_i')$ ）を最上位ビットとする合計10ビットが復号途中結果D612（復号途中結果 $w$ ）として出力される。

#### 【0421】

以上のように、計算器612<sub>i</sub>では、式（9）の演算が行われ、復号途中結果 $w$ が求められる。

#### 【0422】

図30は、図26の計算部615の計算器615<sub>i</sub>の構成例を示すブロック図である。

#### 【0423】

なお、図30では、計算器615<sub>i</sub>について説明するが、計算器615<sub>2</sub>乃至計算器615<sub>5</sub>も同様に構成される。

#### 【0424】

また、図30では、計算器612による第1の演算の結果得られる各復号途中結果（ $w$ ）が符号ビットを合わせて合計10ビットに量子化され、復号途中結果格納用メモリ610から供給される、前回の第2の演算の結果得られた各復号途中結果（ $u_j$ ）が符号ビットを合わせて6ビットに量子化されているものとして、計算器615<sub>i</sub>を表している。さらに、図30の計算器615<sub>i</sub>には、クロック $ck$ が供給され、このクロック $ck$ は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック $ck$ に同期して処理を行う。

#### 【0425】

図30の計算器615<sub>i</sub>は、制御部617から供給される制御信号D622に基づいて、受信メモリ616から1つずつ読み込まれる受信データD617（受信値 $u_{0i}$ ）、サイクリックシフト回路614から1つずつ読み込まれる復号途中結果D614（ $w$ ）、および復号途中結果格納用メモリ610から1つずつ読み込まれる前回の計算部615による第2の演算の結果得られた復号途中結果D610（ $v_i'$ ）とを用いて、式（1）、式（10）、式（11）にしたがう第2の演算を行う。

#### 【0426】

即ち、計算器615<sub>i</sub>では、サイクリックシフト回路614から、検査行列の行に対応する10ビットの復号途中結果D614（復号途中結果 $w$ ）が1つずつ読み込まれとともに、復号途中結果格納用メモリ610から、前回の計算部615による第2の演算の結果得られた6ビットの復号途中結果D610（復号途中結果 $v_i'$ ）が1つずつ読み込まれ、その復号途中結果D614の最上位ビットの符号ビットD651（ $\text{sign}(w)$ ）と復号途中結果D610の最上位ビットの符号ビットD653（ $\text{sign}(u_j)$ ）が、EXOR回路653に供給されるとともに、その復号途中結果D614の下位9ビットの絶対値D652（ $|w|$ ）と復号途中結果D610の下位9ビットの符号ビットD653（ $|v_i'|$ ）が、減算器651に供給される。また、計算器615<sub>i</sub>では、受信メモリ616から6ビットの受信データD617が1つずつ読み込まれ、加算器658に供給される。さらに、計算器615<sub>i</sub>には、制御部617から制御信号D622が供給され、その制御信号D622は、セクタ656に供給される。

#### 【0427】

減算器651は、絶対値D652から絶対値D654を減算し、その5ビットの減算値D655をLUT652に供給する。LUT652は、その減算値D655に対して、 $\phi^{-1}$ の演算を行った5ビットの演算結果D656（ $\phi^{-1}(|w| - |v_i'|)$ ）を出力する。

#### 【0428】

一方、EXOR回路653は、符号ビットD651（ $\text{sign}(w)$ ）と符号ビットD653（ $\text{sign}(v_i')$ ）



))との排他的論理和を演算することにより、符号ビットD651と符号ビットD653を乗算し、1ビットの乗結果を乗算値D657として出力する。そして、LUT652から供給される5ビットの演算結果D656を下位5ビット( $\phi^{-1}(|w| - |v_i'|)$ )とするとともに、EXOR回路653から供給される1ビットの値D657( $\text{sign}(w) \times \text{sign}(v_i')$ )を最上位ビットとした6ビットの値D658が、加算器654に供給されるとともに、FIFOメモリ659に供給される。

#### 【0429】

以上のように、式(10)にしたがう演算が行われ、その演算の結果である6ビットの値D658( $u_j$ )が、加算器654に供給されるとともに、FIFOメモリ659に供給される。

#### 【0430】

加算器654は、6ビットの値D658( $u_j$ )とレジスタ655に格納されている9ビットの値D659とを加算することにより、値D658を積算し、その結果得られる9ビットの積算値を、レジスタ655に再格納する。なお、検査行列の1列に亘る全ての1に対応する値D658が積算された場合、レジスタ655はリセットされる。

#### 【0431】

検査行列の1列に亘る値D658が1つずつ読み込まれ、レジスタ655に1列分の値D658が積算された値が格納された場合、制御部617から供給される制御信号D622は、「0」から「1」に変化する。例えば、列の重みが「5」である場合、制御信号D622は、1から4クロック目までは「0」となり、5クロック目では「1」となる。

#### 【0432】

制御信号D622が「1」の場合、セクタ656は、レジスタ655に格納されている値、即ち、検査行列の1列に亘る1に対応する値D658( $u_j$ )が積算された9ビットの値D659( $j=1$ から $d_v$ までの $\sum u_j$ )を選択し、レジスタ657に出力して格納させる。レジスタ657は、格納している値D659を、9ビットの値D660として、セクタ471と加算器658に供給する。制御信号D622が「0」の場合、セクタ656は、レジスタ657から供給された値D660を選択し、レジスタ657に出力し再格納させる。即ち、検査行列の1列に亘る1に対応する値D658( $u_j$ )が積算されるまで、レジスタ657は、前回積算された値を、セクタ656と加算器658に供給する。

#### 【0433】

加算器658は、9ビットの値D660と、受信用メモリ616から供給された6ビットの受信データD617とを加算して、その結果得られる9ビットの値D661を供給する。

#### 【0434】

計算器615では、最後の演算を行う場合、加算器658が、9ビットの値D661を最終的な復号結果として出力する。即ち、計算部615は、式(5)にしたがって演算を行う。

#### 【0435】

一方、FIFOメモリ659は、レジスタ665から新たな値D660( $j=1$ から $j=d_v$ までの $\sum u_j$ )が出力されるまでの間、6ビットの値D658( $u_j$ )を遅延し、6ビットの値D662として減算器660に供給する。減算器660は、9ビットの値D660から6ビットの値D662を減算し、その減算値D663を出力する。即ち、減算器660は、検査行列の1列に亘る1に対応する値D658の積算値から、求めたい枝に対応する値、即ち検査行列の所定の1に対応する値D658( $u_j$ )を減算して、その減算値( $i=1$ から $i=d_v-1$ までの $\sum u_j$ )を6ビットの減算値D663として出力する。

#### 【0436】

以上のように、式(1)にしたがう演算が行われ、その演算の結果である6ビットの減算値D663( $v_i$ )が出力される。そして、減算器660から出力された6ビットの減算値D663の下位5ビットの絶対値( $|v_i|$ )が、LUT661に供給されるとともに、最上位ビットの符号ビット( $\text{sign}(v_i)$ )が値D665として出力される。

#### 【0437】

LUT661は、絶対値( $|v_i|$ )に対して、 $\phi$ の演算を行った5ビットの演算結果D666(

$\phi(|v_i|)$ )) を出力する。そして、LUT 661 から出力された 5 ビットの演算結果 D666 ( $\phi(|v_i|)$ )) を下位 5 ビットとするとともに、値 D665 ( $\text{sign}(v_i)$ )) を最上位ビットとした合計 6 ビットを、復号途中結果 ( $v_i'$ ) として復号途中結果格納用メモリ 610 に供給する。

#### 【0438】

以上のように、計算器 615<sub>1</sub> では、式 (1)、式 (10)、式 (11) の演算が行われ、復号途中結果  $v_i'$  が求められる。

#### 【0439】

なお、図 15 の検査行列の列の重みの最大は 5 であるため、即ち、計算器 615<sub>1</sub> に供給される復号途中結果 D614 ( $w$ ) と復号途中結果 D610 ( $v_i'$ ) の最大数は 5 であるため、計算器 615<sub>1</sub> は、5 個の復号途中結果 D614 と復号途中結果 D610 から求められる 5 個の演算結果 D658 ( $u_j$ ) を遅延させる FIFO メモリ 659 を有している。列の重みが 5 未満の行のメッセージを計算するときには、FIFO メモリ 659 における遅延量が、その列の重みの値に減らされる。

#### 【0440】

図 31 は、図 26 の復号途中結果格納用メモリ 610 の構成例を示すブロック図である。

#### 【0441】

復号途中結果格納用メモリ 610 は、スイッチ 701 と 704、および 2 つのシングルポート RAM である復号途中結果格納用 RAM 702 と 703 から構成される。

#### 【0442】

この復号途中結果格納用メモリ 610 の各部について詳細に説明する前に、まず、復号途中結果格納用 RAM 702 と 703 へのデータの格納方法について説明する。

#### 【0443】

復号途中結果格納用 RAM 702 と 703 は、計算部 612 による第 1 の演算の結果得られ、スイッチ 701 を介して供給された復号途中結果 D615 を格納する。

#### 【0444】

具体的には、復号途中結果格納用 RAM 702 の第 1 アドレスから第 5 アドレスには、図 15 の検査行列 H の第 1 列目から第 5 列目までの 1 に対応する復号途中結果 D615 (D701) が、各行ともに横方向 (列方向) に詰めた形に (0 を無視した形で) 格納される。

#### 【0445】

即ち、第  $j$  行第  $i$  列を、 $(j, i)$  と表すこととすると、復号途中結果格納用 RAM 702 の第 1 アドレスには、図 15 の検査行列の (1, 1) から (5, 5) の  $5 \times 5$  の単位行列の 1 に対応するデータが、第 2 アドレスには、図 15 の検査行列の (6, 1) から (10, 5) の和行列 ( $5 \times 5$  の単位行列を右に 1 つだけサイクリックシフトした第 1 のシフト行列と、右に 2 つだけサイクリックシフトした第 2 のシフト行列との和である和行列) を構成する第 1 のシフト行列の 1 の位置に対応するデータが格納される。また、第 3 アドレスには、検査行列の (6, 1) から (10, 5) の和行列を構成する第 2 のシフト行列の 1 の位置に対応するデータが格納される。以下、第 4 アドレスおよび第 5 アドレスについても、図 15 の検査行列に対応づけて、データが格納される。

#### 【0446】

復号途中格納用 RAM 702 の第 6 アドレスから第 10 アドレスには、図 15 の検査行列の第 11 列目から第 15 列目までの 1 に対応するデータが格納される。即ち、第 6 アドレスには、検査行列の (11, 11) から (15, 15) の和行列 ( $5 \times 5$  の単位行列と、 $5 \times 5$  の単位行列を右に 3 つだけサイクリックシフトした第 1 のシフト行列との和である和行列) を構成する第 1 のシフト行列の 1 の位置に対応するデータが格納され、第 7 アドレスには、検査行列の (11, 11) から (15, 15) の和行列を構成する単位行列の 1 に対応するデータが格納される。以下、第 8 アドレスから第 10 アドレスについても、検査行列に対応づけてデータが格納される。

#### 【0447】

同様に、復号途中格納用RAM 702の第10アドレスから第28アドレスには、図15の検査行列に対応づけて、第21列目から第25列目まで、第31列目から第35列目まで、第41列目から第45列目まで、第51列目から第55列目まで、第61列目から第65列目まで、第71列目から第75列目まで、第81列目から第85列目までの1に対応するデータが格納される。即ち、復号途中結果格納用RAM 702のワード数は、28である。

**【0448】**

復号途中結果格納用RAM 703の第1アドレスから第5アドレスには、図15の検査行列Hの第6列目から第10列目までの1に対応する復号途中結果D615(D702)が、各行ともに横方向(列方向)に詰めた形に(0を無視した形で)格納される。

**【0449】**

即ち、復号途中結果格納用RAM 703の第1アドレスには、検査行列の構成行列である(6,1)から(10,5)の和行列(5×5の単位行列を右に1つだけサイクリックシフトした第1のシフト行列と、右に2つだけサイクリックシフトした第2のシフト行列の和である和行列)を構成する第1のシフト行列の1に対応するデータが、第2アドレスには、検査行列の構成行列である(6,1)から(10,5)の和行列を構成する第2のシフト行列の1に対応するデータが格納される。以下、第3アドレスから第5アドレスも同様に検査行列の構成行列と対応づけてデータが格納される。

**【0450】**

同様に、復号途中格納用RAM 703の第6アドレスから第26アドレスには、図15の検査行列の第16列目から第20列目まで、第26列目から第30列目まで、第36列目から第40列目まで、第46列目から第50列目まで、第56列目から第60列目まで、第66列目から第70列目まで、第76列目から第80列目まで、第86列目から第90列目までの1に対応するデータが、図15の検査行列に対応づけて格納される。即ち、復号途中結果格納用RAM 703のワード数は、26である。

**【0451】**

上述したように、復号途中結果格納用RAM 702のワード(word)数は、28であり、復号途中結果格納用RAM 703のワード数は、26である。

**【0452】**

図32は、復号途中結果格納用メモリ610の復号途中結果格納用RAM 702と復号途中結果格納用RAM 703の読み出しと書き込みの動作を説明するタイミングチャートである。

**【0453】**

なお、図32において、横軸は、時間(t)を表している。

**【0454】**

復号途中結果格納用メモリ610では、計算部612による第1の演算が行われる場合、制御部617から供給される制御信号D720<sub>2</sub>に基づいて、復号途中結果格納用RAM 702から既に格納されている第2の演算の結果得られる復号途中結果D703を読み出すか、あるいは制御部617から供給される制御信号D720<sub>3</sub>に基づいて、復号途中結果格納用RAM 703から、既に格納されている第2の演算の結果得られる復号途中結果D704を読み出し、その読み出した復号途中結果をスイッチ704を介して、サイクリックシフト回路614に供給する。

**【0455】**

復号途中結果格納用メモリ610には、計算部615により第2の演算が行われる場合、計算部615から第2の演算の結果得られる復号途中結果D615(v<sub>i</sub>')が供給され、その復号途中結果D615が復号途中結果格納用RAM 702または復号途中結果格納用RAM 703のうちの一方の所定のアドレスに書き込まれると同時に、他方から、前回の計算部615による第2の演算の結果得られた復号途中結果D610(v<sub>i</sub>')が読み出され、サイクリックシフト回路614を介して、計算部615に出力される。

**【0456】**

スイッチ701には、計算部615から5つの復号途中結果D615が供給されるとともに、その復号途中結果D615を書き込むメモリとして、復号途中結果格納用RAM702または復号途中結果格納用RAM703の一方の選択を表す制御信号D720<sub>1</sub>が供給される。スイッチ701は、制御信号D720<sub>1</sub>に基づいて、復号途中結果格納用RAM702または復号途中結果格納用RAM703の一方を選択し、その選択した一方に、5つの復号途中結果D612を供給する。

#### 【0457】

復号途中結果格納用RAM702には、スイッチ701から5つの復号途中結果D612が、復号途中結果D701として供給されるとともに、制御部617からアドレスを表す制御信号D720<sub>2</sub>が供給される。復号途中結果格納用RAM702は、制御信号D720<sub>2</sub>が表すアドレスに既に格納されている前回の計算部615による第2の演算の結果得られた5つの復号途中結果D701を読み出し、復号途中結果D703としてスイッチ704に供給する。また、復号途中結果格納用RAM702は、制御信号D720<sub>2</sub>が表すアドレスに、スイッチ701から供給された5つの復号途中結果D702を格納する(書き込む)。

#### 【0458】

復号途中結果格納用RAM703には、スイッチ701から5つの復号途中結果D615が、復号途中結果D702として供給されるとともに、制御部617からアドレスを表す制御信号D720<sub>3</sub>が供給される。復号途中結果格納用RAM703は、制御信号D720<sub>3</sub>が表すアドレスに既に格納されている前回の計算部615による第2の演算の結果得られた5つの復号途中結果D702を読み出し、復号途中結果D704としてスイッチ704に供給する。また、復号途中結果格納用RAM702は、制御信号D720<sub>3</sub>が表すアドレスに、スイッチ701から供給された5つの復号途中結果D702を格納する(書き込む)。

#### 【0459】

スイッチ704には、復号途中結果格納用RAM702から復号途中結果D703が供給されるか、あるいは復号途中結果格納用RAM703から復号途中結果D704が供給される。また、制御部617から、復号途中結果格納用RAM702または復号途中結果格納用RAM703の一方の選択を表す制御信号D720<sub>4</sub>が供給される。スイッチ704は、制御信号D720<sub>4</sub>に基づいて、復号途中結果格納用RAM702または復号途中結果格納用RAM703の一方を選択し、その選択した一方から供給された5つの復号途中結果を、5つの復号途中結果D610として計算部615に供給する。

#### 【0460】

復号途中結果格納用メモリ610では、計算部615により第2の演算が行われる場合、復号途中結果格納用RAM702が、制御信号D720<sub>2</sub>に基づいて、既に格納している、前回の計算部615の第2の演算の結果得られた復号途中結果D701のうち、同一アドレスに格納している検査行列の第1列目から第5列目までの1に対応する復号途中結果D701を、5つ単位で、5回読み出し、スイッチ704を介して、計算部615に供給する。即ち、図15の検査行列Hの列重みは、5であるため、検査行列Hの各列の1に対応する復号途中結果は5つあり、復号途中結果格納用RAM702は、第1行列目から第5列目までの1に対応する復号途中結果D701を、5つ単位で5回読み出す。

#### 【0461】

次に、復号途中結果格納用RAM703は、制御信号D720<sub>3</sub>に基づいて、既に格納している、前回の計算部615による第2の演算の結果得られた復号途中結果D702のうち、同一アドレスに格納している検査行列の第6列目から第10列目までの1に対応する5つの復号途中結果D702を、5回続けて読み出し、スイッチ704とサイクリックシフト回路614を介して、計算部615に供給する。それと同時に、復号途中結果格納用RAM702には、計算部615により、いま行われている第2の演算の結果得られる検査行列の第1列目から第5列目までの1に対応する5つの復号途中結果D615がスイッチ701を介して、復号途中結果D701として供給され、復号途中結果格納用RAM702は、その復号途中結果D701を、制御信号D720<sub>2</sub>に基づいて、既に読み出された復号途中結果D703が格納されていたアドレスに5回続けて格納する。

## 【0462】

その後、復号途中結果格納用RAM 702は、制御信号D720<sub>2</sub>に基づいて、既に格納している、前回の計算部615による第2の演算の結果得られた復号途中結果D701のうち、同一アドレスに格納している検査行列の第11列目から第15列目までの1に対応する復号途中結果D701を、5つ単位で5回続けて読み出し、スイッチ704を介して、計算部615に供給する。それと同時に、復号途中結果格納用RAM 703には、計算部615により、いま行われている第2の演算の結果得られる検査行列の第6列目から第10列目までの1に対応する5つの復号途中結果D612がスイッチ701を介して、復号途中結果D702として供給され、復号途中結果格納用RAM 703は、その復号途中結果D702を、制御信号D720<sub>3</sub>に基づいて、既に読み出された復号途中結果D704が格納されていたアドレスに5回続けて格納する。

## 【0463】

以後、同様に、計算部615による第2の演算の結果得られる検査行列の全ての1に対応する復号途中結果が、復号途中結果格納用RAM 702または復号途中結果格納用RAM 703に格納されるまで、復号途中結果格納用RAM 702と復号途中結果格納用RAM 703は、5回ずつの読み出しまたは書き込みを交互に行う。

## 【0464】

図33は、図26の復号装置600の復号処理を説明するフローチャートである。この処理は、例えば、受信用メモリ616に復号すべき受信データが格納されたとき、開始される。

## 【0465】

ステップS70において、サイクリックシフト回路614は、復号途中結果格納用メモリ613から供給された後述するステップS76で格納された5つの復号途中結果D613を、並べ替えてサイクリックシフトを行い、計算部615に供給する。

## 【0466】

具体的には、サイクリックシフト回路614には、復号途中結果格納用メモリ613から5つの復号途中結果D613が供給されるとともに、制御部617から、その復号途中結果D613に対応する検査行列の1が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報（Matrixデータ）を表す制御信号D621が供給される。サイクリックシフト回路614は、制御信号D621を元に、5つの復号途中結果D613をサイクリックシフトし（並べ替え）、その結果を復号途中結果D614として、計算部615に供給する。

## 【0467】

なお、受信用メモリ616から供給された受信データD617に対して、まだ第1の演算が行われておらず、復号途中結果格納用メモリ613に復号途中結果D612が格納されていない場合、計算部615は、初期値に設定する。

## 【0468】

ステップS71において、計算部615は、第2の演算を行い、その演算の結果である復号途中結果D615を復号途中結果格納用メモリ610に供給する。

## 【0469】

具体的には、計算部615には、ステップS70でサイクリックシフト回路614から5つの復号途中結果D614が供給されるとともに、直前の後述するステップS72で復号途中結果格納用メモリ610から復号途中結果D610が供給される。また、受信データ用メモリ616から5つの受信データD617が供給され、5つの復号途中結果D615とD610、受信データD617が、計算部615の計算器615<sub>1</sub>乃至615<sub>5</sub>のそれぞれに1つずつ供給される。さらに、計算部615には、制御部617から制御信号D622が供給され、その制御信号D622が計算器615<sub>1</sub>乃至615<sub>5</sub>に供給される。

## 【0470】

計算器615<sub>1</sub>乃至615<sub>5</sub>は、復号途中結果D614とD610と、受信データD617とを用いて、制御信号D622に基づいて、式(1)、式(10)、式(11)にしたがって、それぞれ

演算を行い、その演算の結果得られる検査行列の各列の 1 に対応する復号途中結果 D615( $v_i'$ )を復号途中結果格納用メモリ 610 に供給する。

**【0471】**

ステップ S71 の処理後は、ステップ S72 に進み、復号途中結果格納用メモリ 610 は、ステップ S71 で計算部 615 から供給された復号途中結果 D615 を、同一アドレスに格納するとともに、既に格納している復号途中結果 D615 (D610) を読み出して、サイクリックシフト回路 611 と計算部 615 に供給する。

**【0472】**

ステップ S72 の処理後は、ステップ S73 に進み、制御部 617 は、計算部 615 により、検査行列の各列の 1 に対応する全ての復号途中結果 D615 が演算されたかどうかを判定し、全ての復号途中結果 D615 が演算されていないと判定した場合、ステップ S70 に戻り、上述した処理を繰り返す。

**【0473】**

一方、ステップ S73 において、制御部 617 は、計算部 615 により、全ての復号途中結果 D615 が演算されたと判定した場合、ステップ S74 に進み、サイクリックシフト回路 611 は、復号途中結果格納用メモリ 610 から供給される復号途中結果 D610( $v_i'$ )をサイクリックシフトする。

**【0474】**

具体的には、サイクリックシフト回路 611 には、復号途中結果格納用メモリ 610 から 5 つの復号途中結果 D610 が供給されるとともに、制御部 617 から、その復号途中結果 D610 に対応する検査行列の 1 が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報 (Matrix データ) を表す制御信号 D619 が供給される。サイクリックシフト回路 611 は、制御信号 D619 を元に、5 つの復号途中結果 D610 をサイクリックシフトし (並べ替え)、その結果を復号途中結果 D611 として、計算部 612 に供給する。

**【0475】**

ステップ S74 の処理後は、ステップ S75 に進み、計算部 612 は、第 1 の演算を行い、その演算結果である復号途中結果 D612 をサイクリックシフト回路 614 に供給する。

**【0476】**

具体的には、計算部 612 には、ステップ S74 でサイクリックシフト回路 611 から 5 つの復号途中結果 D611( $v_i'$ )が供給され、その復号途中結果 D611 が、計算部 612 の計算器 612<sub>1</sub>乃至 612<sub>5</sub>のそれぞれに 1 つずつ供給される。さらに、計算部 612 には、制御部 617 から制御信号 D621 が供給され、その制御信号 D621 が計算器 612<sub>1</sub>乃至 612<sub>5</sub>に供給される。

**【0477】**

計算器 612<sub>1</sub>乃至 612<sub>5</sub>は、それぞれ復号途中結果 D611 を用いて、制御信号 D619 に基づいて、式 (9) にしたがって、それぞれ演算を行い、その演算の結果得られる検査行列の行に対応する復号途中結果 D612 ( $w$ ) を復号途中結果格納用メモリ 613 に供給する。

**【0478】**

ステップ S75 の処理後は、ステップ S76 に進み、復号途中結果格納用メモリ 613 は、ステップ S75 で計算部 612 から供給された復号途中結果 D612 を、同一アドレスに格納し、ステップ S77 に進む。

**【0479】**

ステップ S77 において、制御部 617 は、計算部 612 により、検査行列の全ての行に対応する復号途中結果 D612 が演算されたかどうかを判定し、全ての復号途中結果が演算されていないと判定した場合、ステップ S74 に戻り、上述した処理を繰り返す。

**【0480】**

一方、ステップ S77 において、制御部 617 は、計算部 612 により、全ての行に対応する復号途中結果 D612 が演算されたと判定した場合、処理を終了する。

**【0481】**

なお、復号装置 600 は、復号回数だけ図 33 の復号処理を繰り返し行ない、計算部 621 により上述した式 (5) にしたが演算の結果得られる値 D661 が、最終的な復号結果として出力される。

#### 【0482】

上述した説明では、復号途中結果格納用メモリ 610 は、2つのシングルポート RAM から構成したが、1つの RAM に対して読み出しと書き込みが同時に起こらないようにすれば、3つ以上の RAM から構成してもよい。例えば、RAM の物理的なビットが足りない場合には、複数の RAM を用いて同じ制御信号を与えることで、論理的に 1つの RAM とみなすことができる。

#### 【0483】

また、枝データ (枝に対応するメッセージ) が欠けている箇所に関しては、メモリ格納時 (復号途中結果格納用メモリ 610 と 613 へのデータ格納時) には、何のメッセージも格納せず、また、演算時 (計算部 612 での第 1 の演算時と計算部 615 での第 2 の演算時) にも何の演算も行わない。

#### 【0484】

また、図 16 のサイクリックシフト回路 314 および 320、図 18 のサイクリックシフト回路 411 および 414、図 26 のサイクリックシフト回路 611 および 614 には、バレルシフタを用いると回路規模を小さくしながら所望の操作を実現できる。

#### 【0485】

上述の場合には、説明を簡単にするために、P が 5 の場合、即ち、検査行列を構成する構成行列の行数および列数が 5 の場合を例に挙げたが、構成行列の行数および列数は必ずしも 5 である必要はなく、検査行列によって異なる値を取ることもあり得る。例えば、P は 360 や 392 であってもよい。

#### 【0486】

また、本実施の形態では、符号長 90、符号化率 2/3 の LDPC 符号を用いたが、LDPC 符号の符号長や符号化率は、幾つであっても構わない。例えば、構成行列の行数および列数 P が 5 の場合、枝総数が 5 以下であれば、どんな符号長、符号化率の LDPC 符号でも、制御信号を代えるだけで、図 16 の復号装置 300、図 18 の復号装置 400、図 26 の復号装置 600 を用いて復号可能である。

#### 【0487】

さらに、構成行列の行数および列数 P が所定の値で、枝の総数がある値以下、という条件を満たすある LDPC 符号の復号装置は、その条件を満たす、任意の符号長で、任意の符号化率の LDPC 符号を復号することができる。

#### 【0488】

検査行列が、構成行列の行数および列数 P の倍数でない場合は、検査行列の端数の外側にすべて 0 (all 0) の成分を付けて P の倍数とみなして適用できることがある。

#### 【0489】

次に、上述した一連の処理は、ハードウェアにより行うこともできるし、ソフトウェアにより行うこともできる。一連の処理をソフトウェアによって行う場合には、そのソフトウェアを構成するプログラムが、汎用のコンピュータ等にインストールされる。

#### 【0490】

そこで、図 34 は、上述した一連の処理を実行するプログラムがインストールされるコンピュータの一実施の形態の構成例を示している。

#### 【0491】

プログラムは、コンピュータに内蔵されている記録媒体としてのハードディスク 905 や ROM 903 に予め記録しておくことができる。

#### 【0492】

あるいはまた、プログラムは、フレキシブルディスク、CD-ROM (Compact Disc Read Only Memory)、MO (Magneto Optical) ディスク、DVD (Digital Versatile Disc)、磁気ディスク、半導体メモリなどのリムーバブル記録媒体 911 に、一時的あるいは永続的に格納 (

記録)しておくことができる。このようなりムーバブル記録媒体 911 は、いわゆるパッケージソフトウェアとして提供することができる。

#### 【0493】

なお、プログラムは、上述したようなりムーバブル記録媒体 911 からコンピュータにインストールする他、ダウンロードサイトから、デジタル衛星放送用の人工衛星を介して、コンピュータに無線で転送したり、LAN(Local Area Network)、インターネットといったネットワークを介して、コンピュータに有線で転送し、コンピュータでは、そのようにして転送されてくるプログラムを、通信部 908 で受信し、内蔵するハードディスク 905 にインストールすることができる。

#### 【0494】

コンピュータは、CPU(Central Processing Unit) 902 を内蔵している。CPU 902 には、バス 901 を介して、入出力インタフェース 910 が接続されており、CPU 902 は、入出力インタフェース 910 を介して、ユーザによって、キーボードや、マウス、マイク等で構成される入力部 907 が操作等されることにより指令が入力されると、それにしたがって、ROM(Read Only Memory) 903 に格納されているプログラムを実行する。あるいは、また、CPU 902 は、ハードディスク 905 に格納されているプログラム、衛星若しくはネットワークから転送され、通信部 908 で受信されてハードディスク 905 にインストールされたプログラム、またはドライブ 909 に装着されたりムーバブル記録媒体 911 から読み出されてハードディスク 905 にインストールされたプログラムを、RAM(Random Access Memory) 904 にロードして実行する。これにより、CPU 902 は、上述したフローチャートにしたがった処理、あるいは上述したブロック図の構成により行われる処理を行う。そして、CPU 902 は、その処理結果を、必要に応じて、例えば、入出力インタフェース 910 を介して、LCD(Liquid Crystal Display)やスピーカ等で構成される出力部 906 から出力、あるいは、通信部 908 から送信、さらには、ハードディスク 905 に記録等させる。

#### 【0495】

ここで、本明細書において、コンピュータに各種の処理を行わせるためのプログラムを記述する処理ステップは、必ずしもフローチャートとして記載された順序に沿って時系列に処理する必要はなく、並列的あるいは個別に実行される処理(例えば、並列処理あるいはオブジェクトによる処理)も含むものである。

#### 【0496】

また、プログラムは、1のコンピュータにより処理されるものであっても良いし、複数のコンピュータによって分散処理されるものであっても良い。さらに、プログラムは、遠方のコンピュータに転送されて実行されるものであっても良い。

#### 【0497】

以上のように、 $P \times P$ の単位行列、そのコンポーネントの1のうち1個以上が0になった準単位行列、単位行列もしくは準単位行列をサイクリックシフトしたシフト行列、単位行列、準単位行列、もしくはシフト行列の複数の和である和行列、 $P \times P$ の0行列の組合せで表わすことができる検査行列を持つLDPC符号の復号を、チェックノードとバリエブルノードの演算をP個同時に行うアーキテクチャ(architecture)を採用することにより、ノードの演算を、P個同時に行うことで動作周波数を実現可能な範囲に抑えることができ、多数の繰り返し復号を行うことを可能にしつつ、メモリ(FIFOやRAM)への書き込みと読み出し時に、異なるアドレスへの同時アクセスが起きることを防止することができる。

#### 【0498】

さらに、図16の復号装置300を繰り返し用いて、図15の検査行列で表わされるLDPC符号を復号する場合には、269個の枝をチェックノード、バリエブルノード毎に5個ずつ演算することが可能であることから、1回の復号に、 $269/5 \times 2 = 108$ クロック動作すればよいことになる。50回の復号には、90個の符号情報を受信する間に、 $108 \times 50 = 5400$ クロック動作すればよいことになり、受信周波数の約60倍の動作周波数でよいことになる。従って、図16の復号装置300によれば、各ノード演算を一つずつ行う図9の



復号装置に比べて、1/5の動作周波数で済むことになる。また、回路規模の面から見ても、メモリの大きさは同じであるため、論理回路が多少大きくなっても全体への影響は小さいと言える。

#### 【0499】

さらに、図18の復号装置400と図26の復号装置600は、図16の復号装置300に比べて、メモリの容量が小さくなっている。

#### 【0500】

例えば、LDPC符号の検査行列が図15の検査行列であり、LDPC符号の量子化ビット数が6ビットである場合、図16の復号装置300では、枝データ格納メモリに、全枝数の $269 \times 6 = 1614$ ビットの容量を有するRAM2つ、即ち、2つのRAMで $1614 \times 2 = 3228$ ビットの容量が必要であった。これに対して、例えば、復号途中結果 $v$ の量子化ビット数が9ビットである場合、図18の復号装置400では、復号途中結果格納用メモリ413に、全枝数の1614ビットの容量を有するRAMと、復号途中結果格納用メモリ410に、LDPC符号の符号長（検査行列の列数）と復号途中結果 $v$ の量子化ビット数との乗算値、即ち $90 \times 9 = 810$ ビットの容量を有するRAMを備えればよく、復号装置の回路規模を小さくすることができる。さらに、図18の復号装置400では、第2の演算を行う計算部415において、FIFOメモリを有する必要がないので、ロジックの回路規模を小さくすることができる。

#### 【0501】

また、例えば、LDPC符号の検査行列が図15の検査行列で、復号途中結果 $v$ の量子化ビット数が10ビットである場合、図26の復号装置600では、復号途中結果格納用メモリ610に、全枝数の1614ビットの容量を有するRAMと、復号途中結果格納用メモリ613に、検査行列の行数と復号途中結果 $v$ との乗算値、即ち $30 \times 10 = 300$ ビットの容量を有するRAMを備えればよく、復号装置の回路規模を小さくすることができる。さらに、図26の復号装置600では、第1の演算を行う計算部612において、FIFOメモリを有する必要がないので、ロジックの回路規模を小さくすることができる。

#### 【0502】

一般的に、LDPC符号は符号長が数千から数万と大きいため、 $P$ の値も数百の大きさを持つものが使われる。その場合には、更に本発明に係る復号装置を用いる効果は大きくなる。

#### 【0503】

また、本発明に係る復号装置は、サムプロダクトアルゴリズムを忠実に実装するものであるため、メッセージの量子化以外の復号損失が起きることではない。

#### 【0504】

以上の観点から、本発明に係る復号装置を用いることで、高性能な復号が可能になる。

#### 【図面の簡単な説明】

#### 【0505】

【図1】 LDPC符号の検査行列 $H$ を説明する図である。

【図2】 LDPC符号の復号手順を説明するフローチャートである。

【図3】 メッセージの流れを説明する図である。

【図4】 LDPC符号の検査行列の例を示す図である。

【図5】 検査行列のタナグラフを示す図である。

【図6】 バリアブルノードを示す図である。

【図7】 チェックノードを示す図である。

【図8】 LDPC符号の検査行列の例を示す図である。

【図9】 ノード演算を一つずつ行うLDPC符号の復号装置の構成例を示すブロック図である。

【図10】 メッセージを一つずつ計算するチェックノード計算器の構成例を示すブロック図である。

【図11】 メッセージを一つずつ計算するバリアブルノード計算器の構成例を示すブ

ロック図である。

【図12】 ノード演算を全て同時に行うLDPC符号の復号装置の構成例を示すブロック図である。

【図13】 メッセージを同時に計算するチェックノード計算器の構成例を示すブロック図である。

【図14】 メッセージを同時に計算するバリエブルノード計算器の構成例を示すブロック図である。

【図15】  $5 \times 5$  単位に分割した検査行列を示す図である。

【図16】 本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

【図17】 図16の復号装置の復号処理を説明するフローチャートである。

【図18】 本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

【図19】 チェックノード計算器の構成例を示すブロック図である。

【図20】 バリエブルノード計算器の構成例を示すブロック図である。

【図21】 図18の計算器の構成例を示すブロック図である。

【図22】 図18の計算器の構成例を示すブロック図である。

【図23】 図18の復号途中結果格納用メモリの構成例を示すブロック図である。

【図24】 図18の復号途中結果格納用RAMの動作を説明するタイミングチャートである。

【図25】 図18の復号装置の復号処理を説明するフローチャートである。

【図26】 本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

【図27】 チェックノード計算器の構成例を示すブロック図である。

【図28】 バリエブルノード計算器の構成例を示すブロック図である。

【図29】 図26の計算器の構成例を示すブロック図である。

【図30】 図26の計算器の構成例を示すブロック図である。

【図31】 図26の復号途中結果格納用メモリの構成例を示すブロック図である。

【図32】 図31の復号途中結果格納用RAMの動作を説明するタイミングチャートである。

【図33】 図26の復号装置の復号処理を説明するフローチャートである。

【図34】 本発明を適用したコンピュータの一実施の形態の構成例を示すブロック図である。

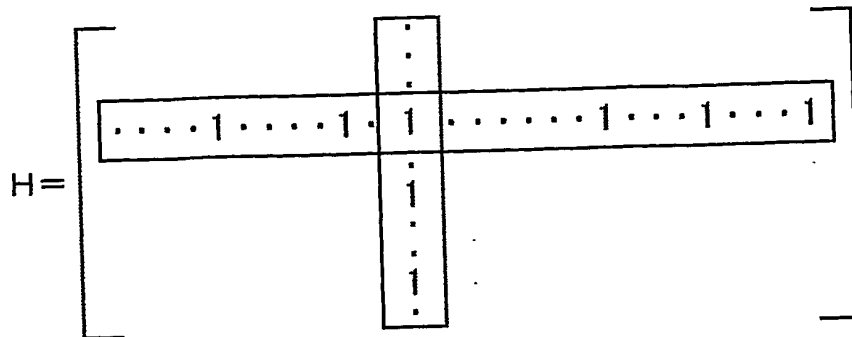
#### 【符号の説明】

##### 【0506】

300 スイッチ, 311 枝データ格納用メモリ, 312 セレクタ, 313  
 チェックノード計算部, 314 サイクリックシフト回路, 315 スイッチ, 3  
 16 枝データ格納用メモリ, 317 セレクタ, 318 受信データ用メモリ,  
 319 バリエブルノード計算部, 320 サイクリックシフト回路, 901 バス  
 , 902 CPU, 903 ROM, 904 RAM, 905 ハードディスク, 90  
 6 出力部, 907 入力部, 908 通信部, 909 ドライブ, 910 入  
 出力インタフェース, 911 リムーバブル記録媒体

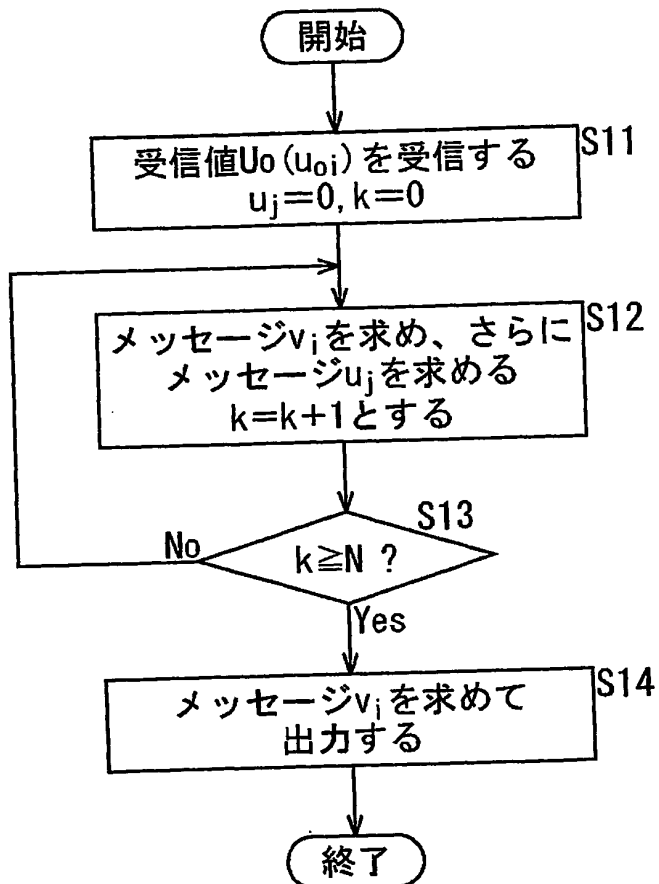
【書類名】 図面  
【図 1】

図1



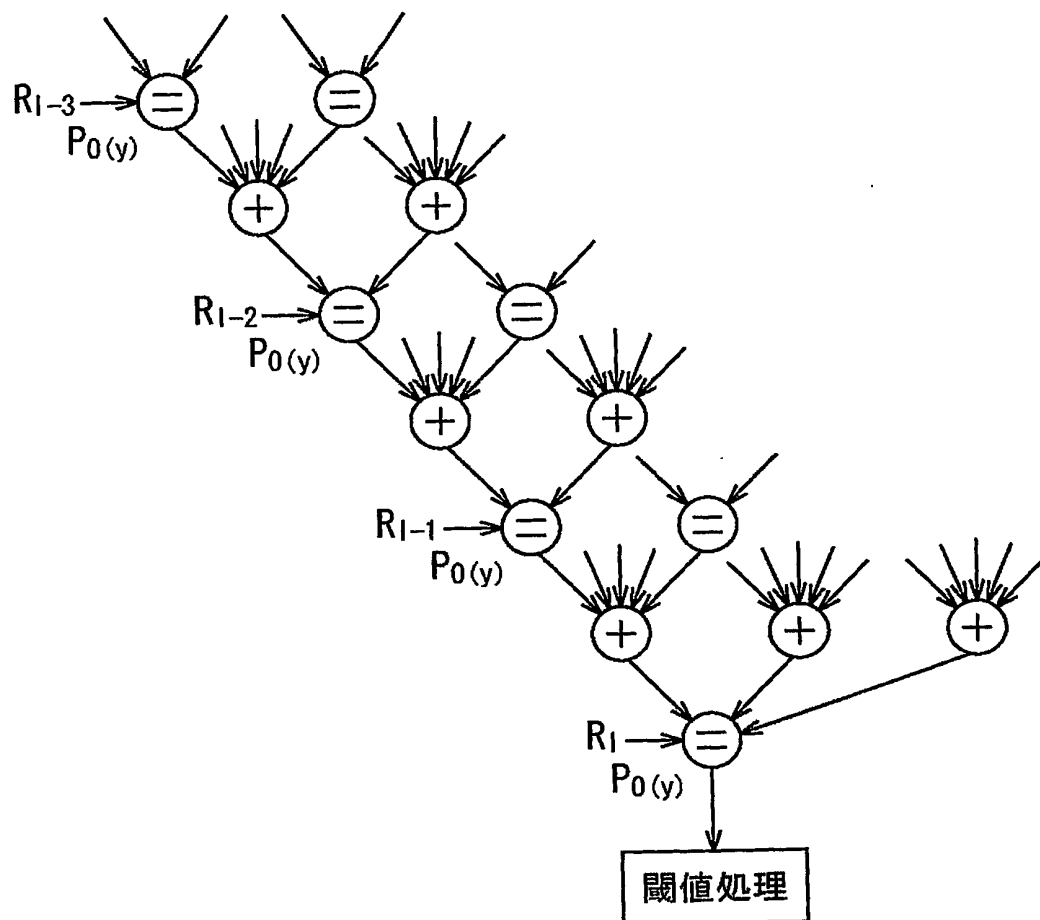
【図 2】

図2



【図 3】

図3

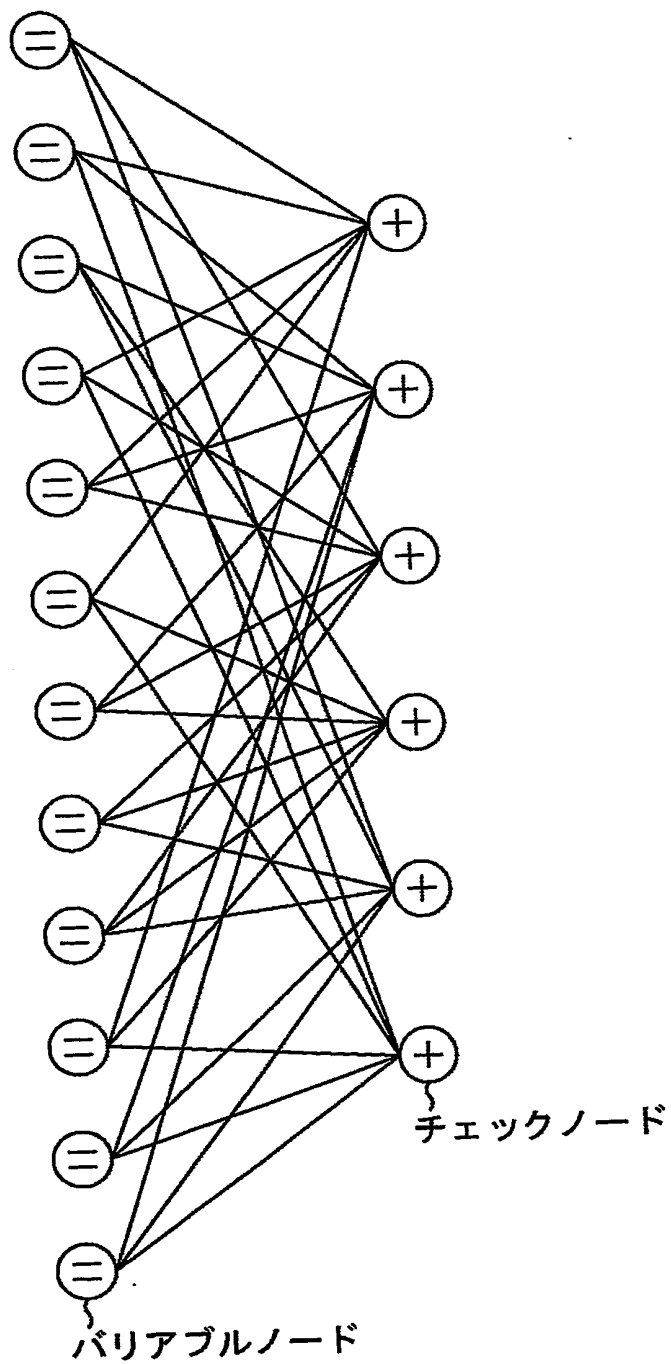


【図 4】

図4

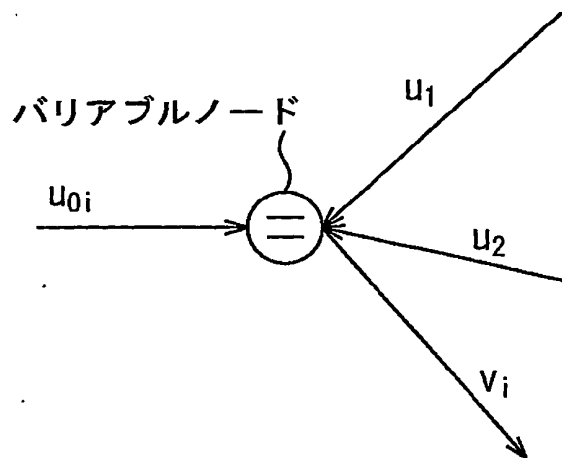
$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

【図5】  
図5



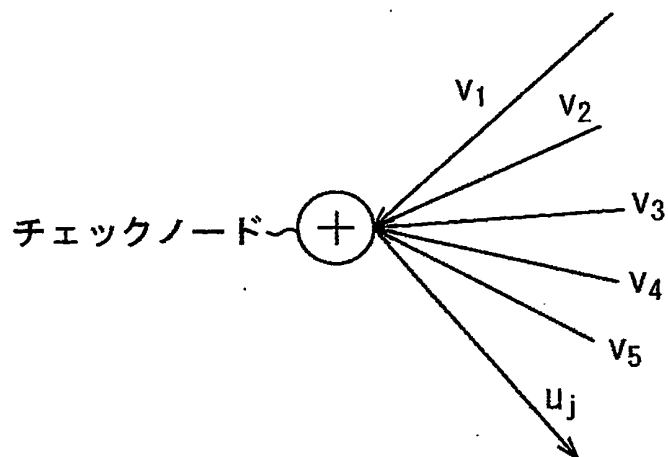
【図6】

図6



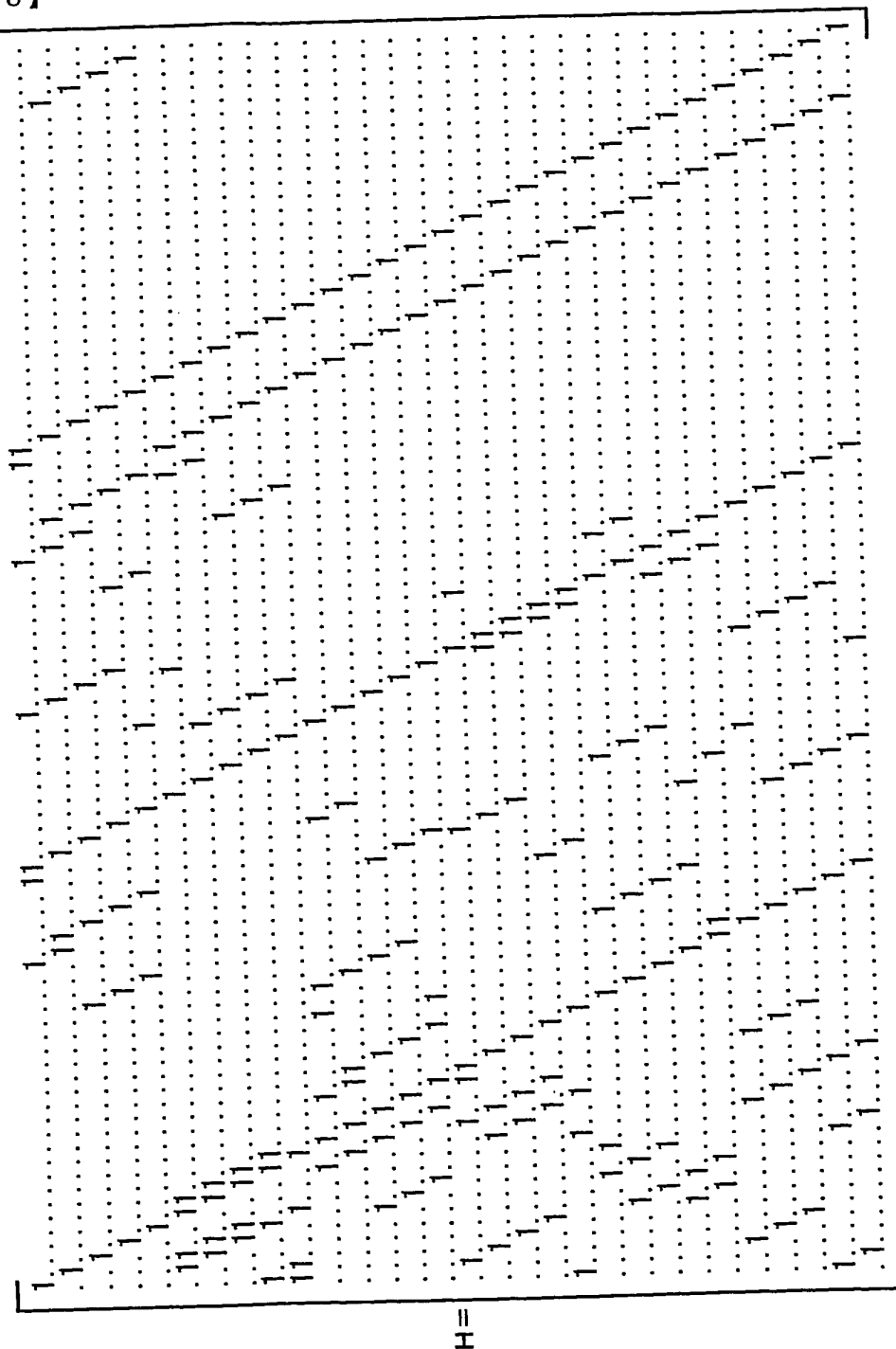
【図7】

図7

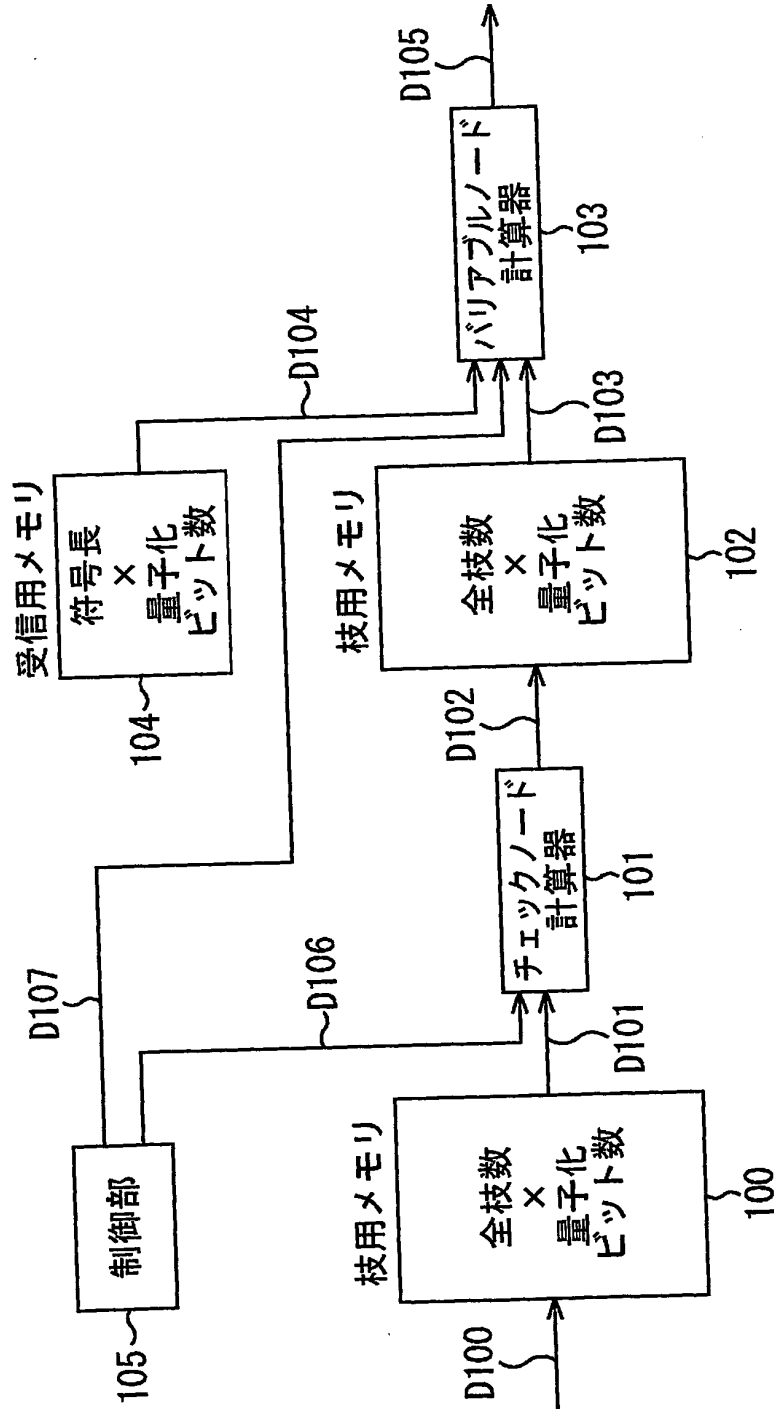


【図 8】

図8



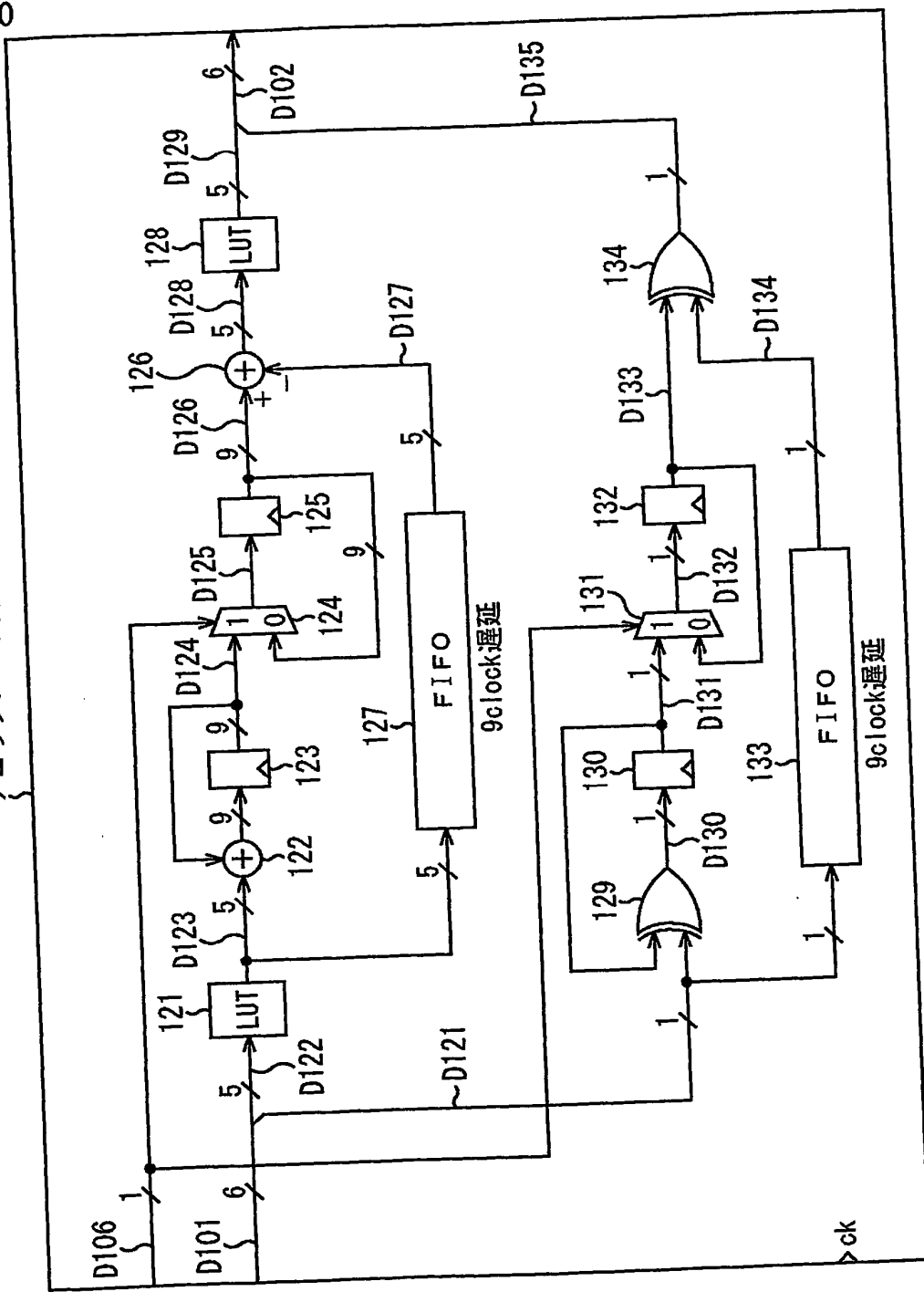
【図9】  
図9





【図10】  
図10

チェックノード計算器 101



【图 1 1】  
图 11

图11

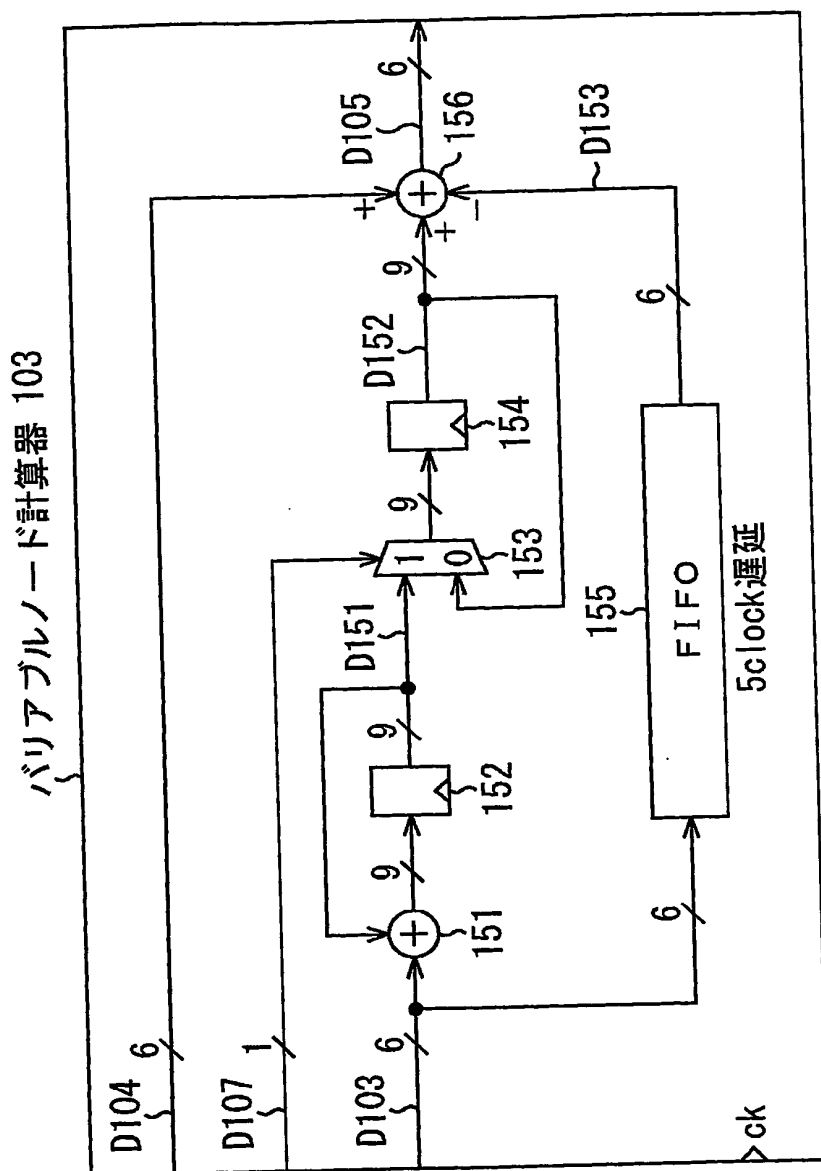
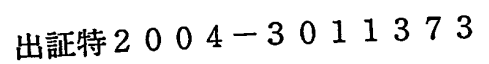
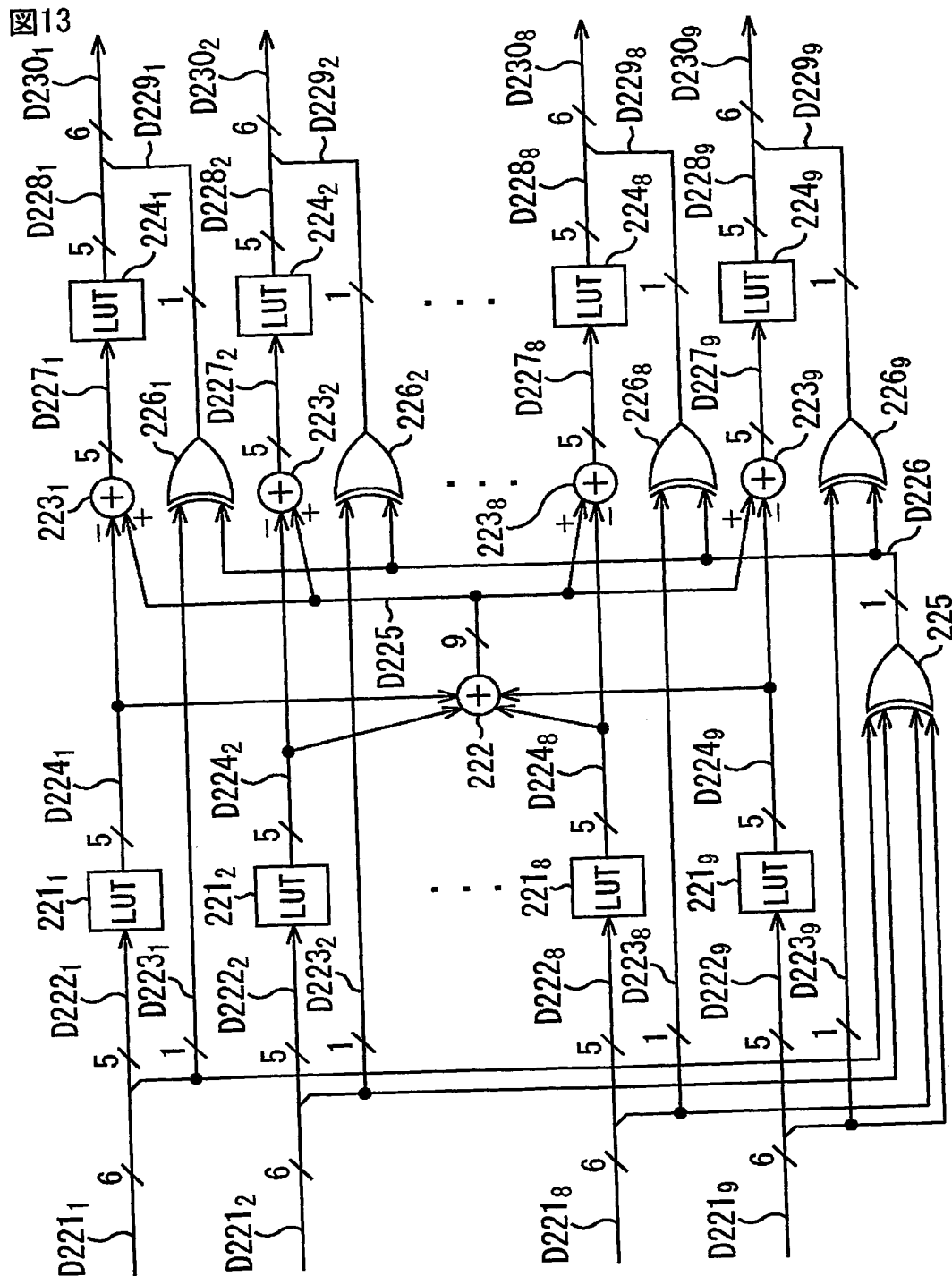


图12

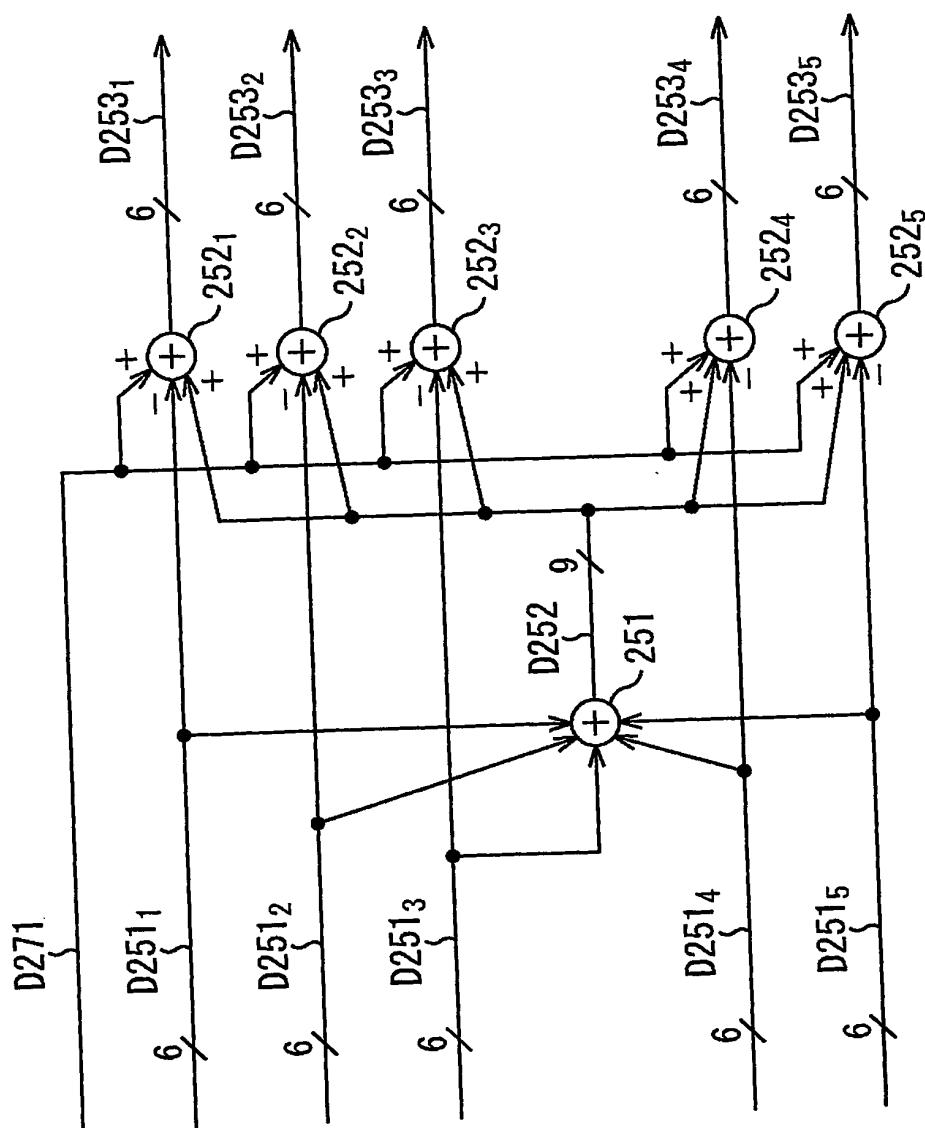


【図 13】

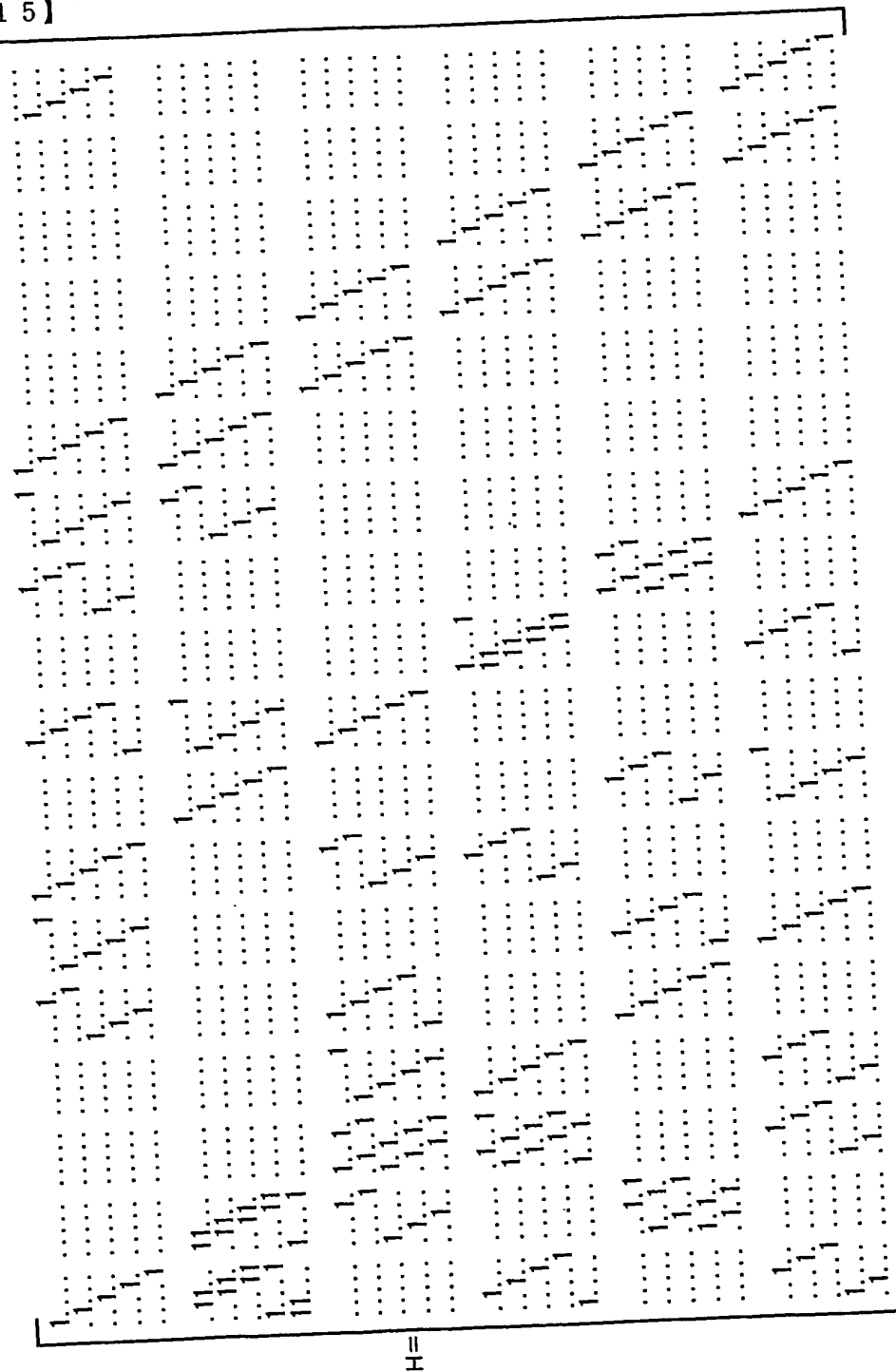


【図 14】

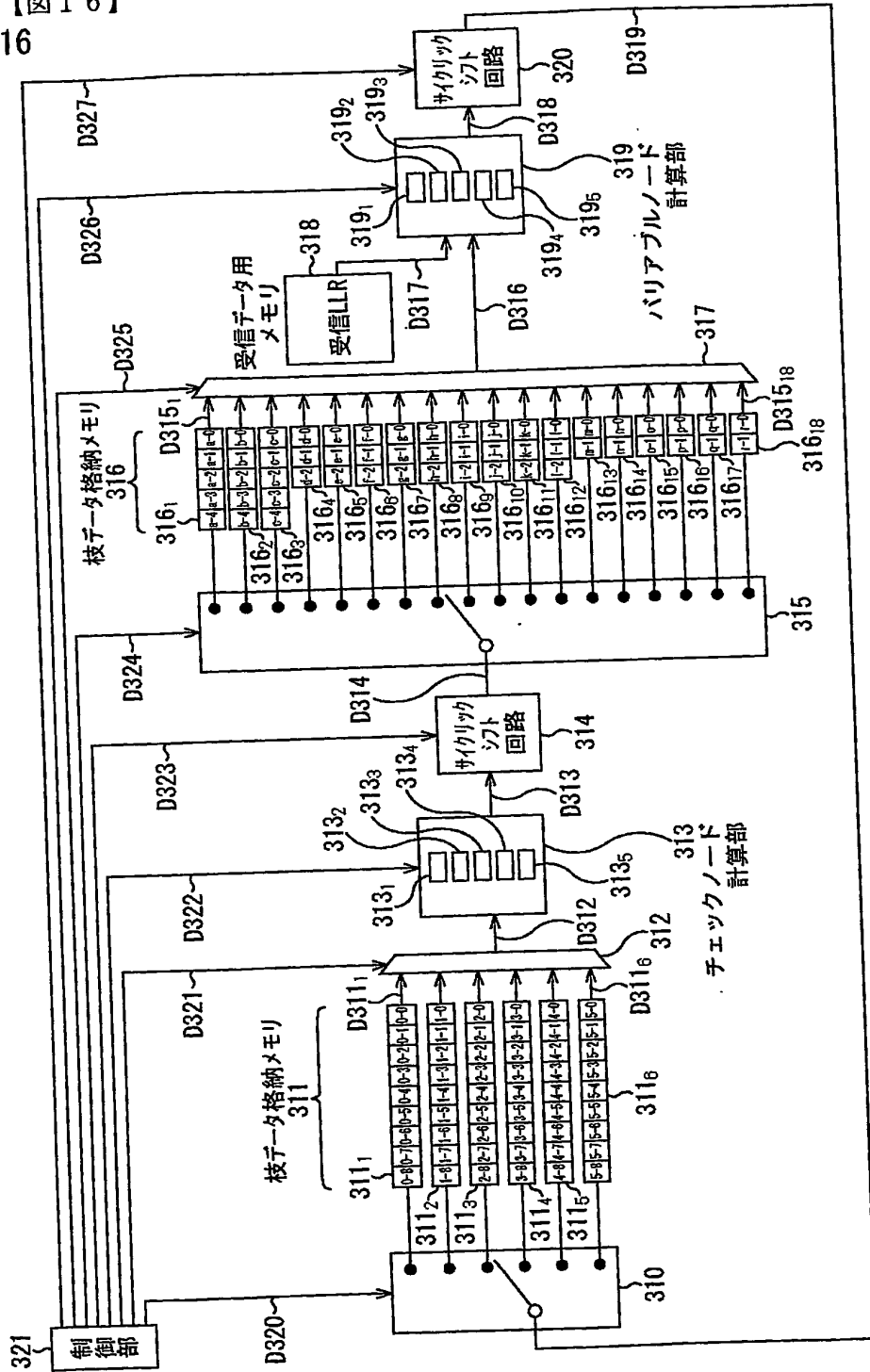
図14



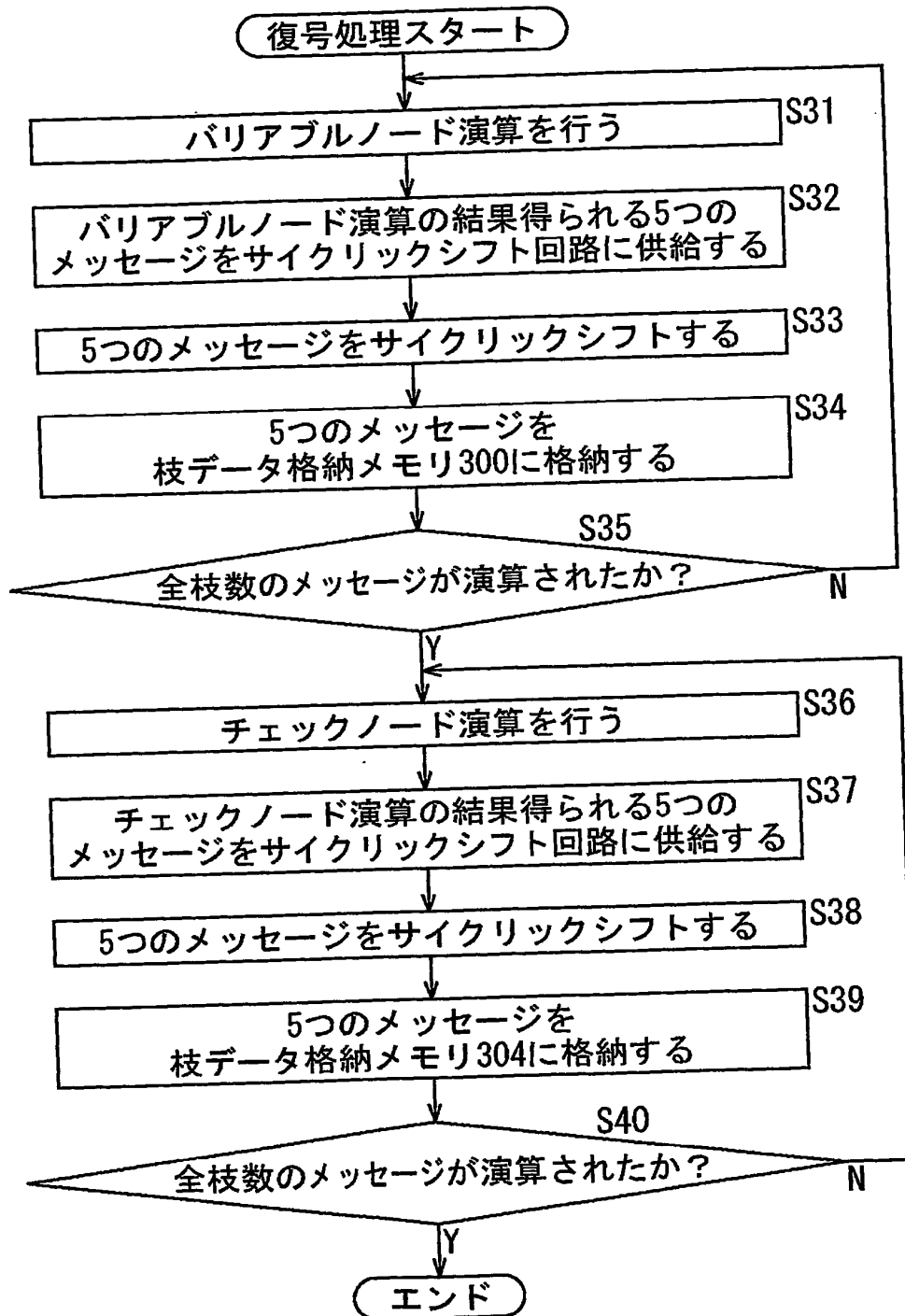
【図 15】  
図15



【図16】  
図16

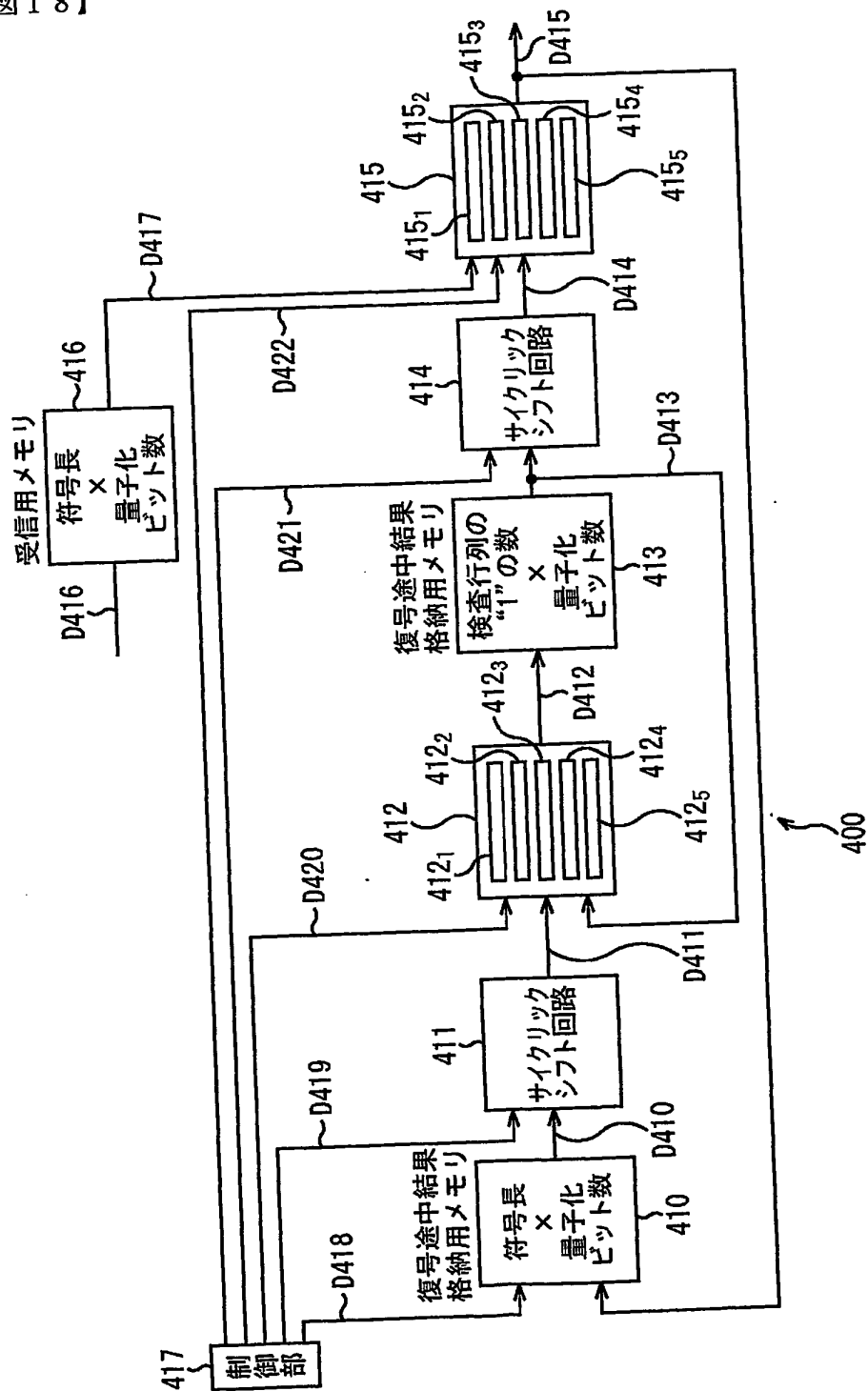


300

【図17】  
図17



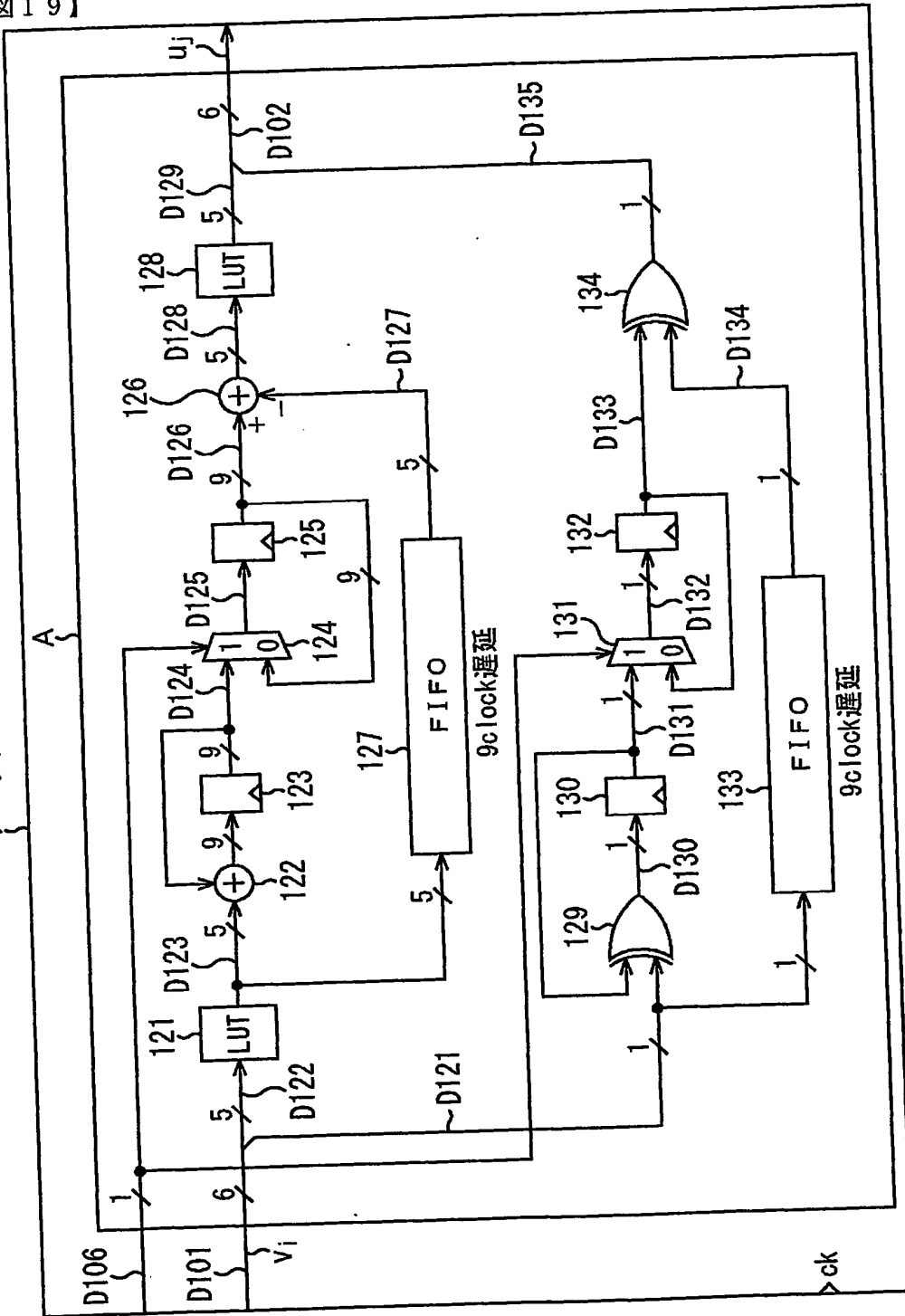
【図18】  
図18



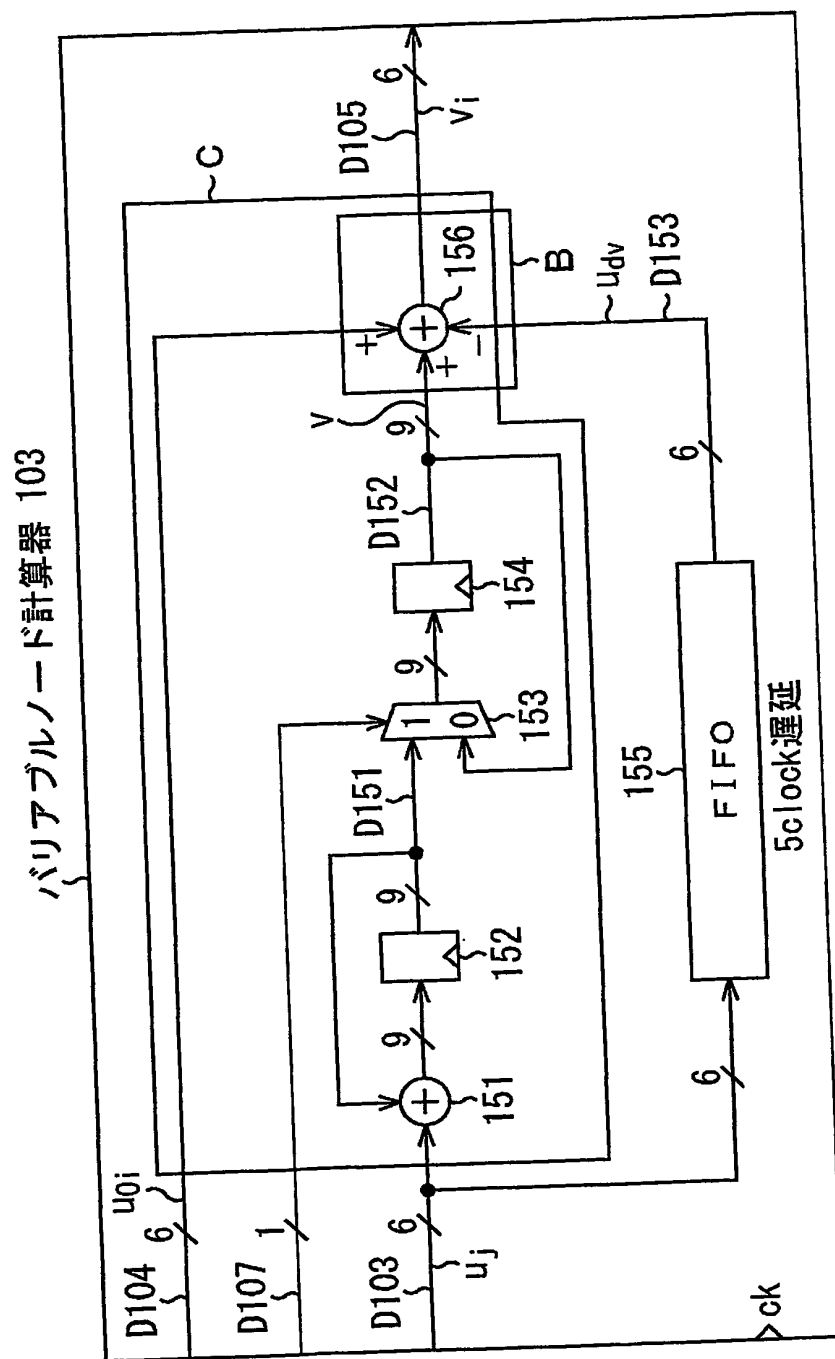
【図19】

図19

チェックノード計算器 101

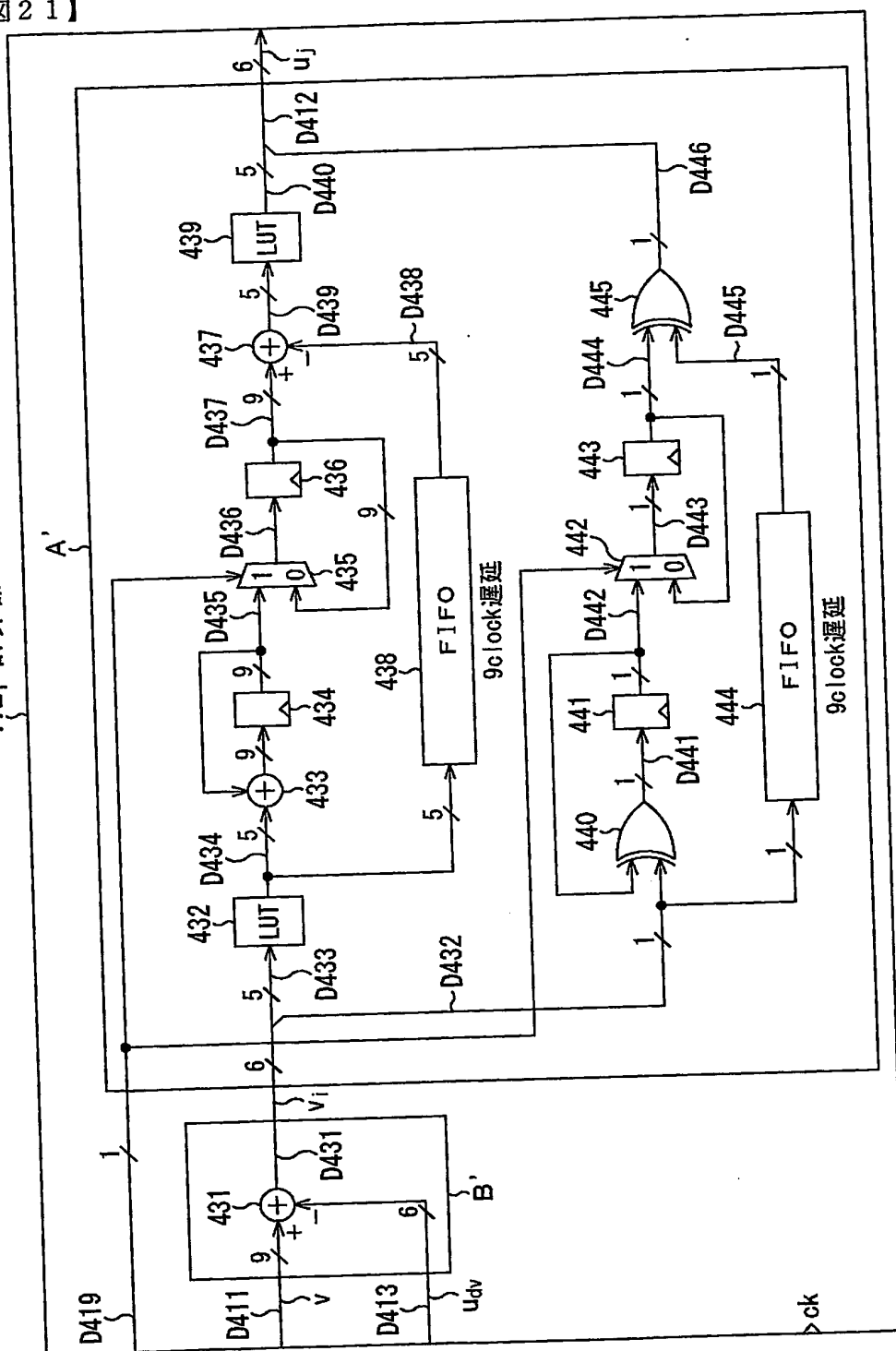


【图 20】  
图 20

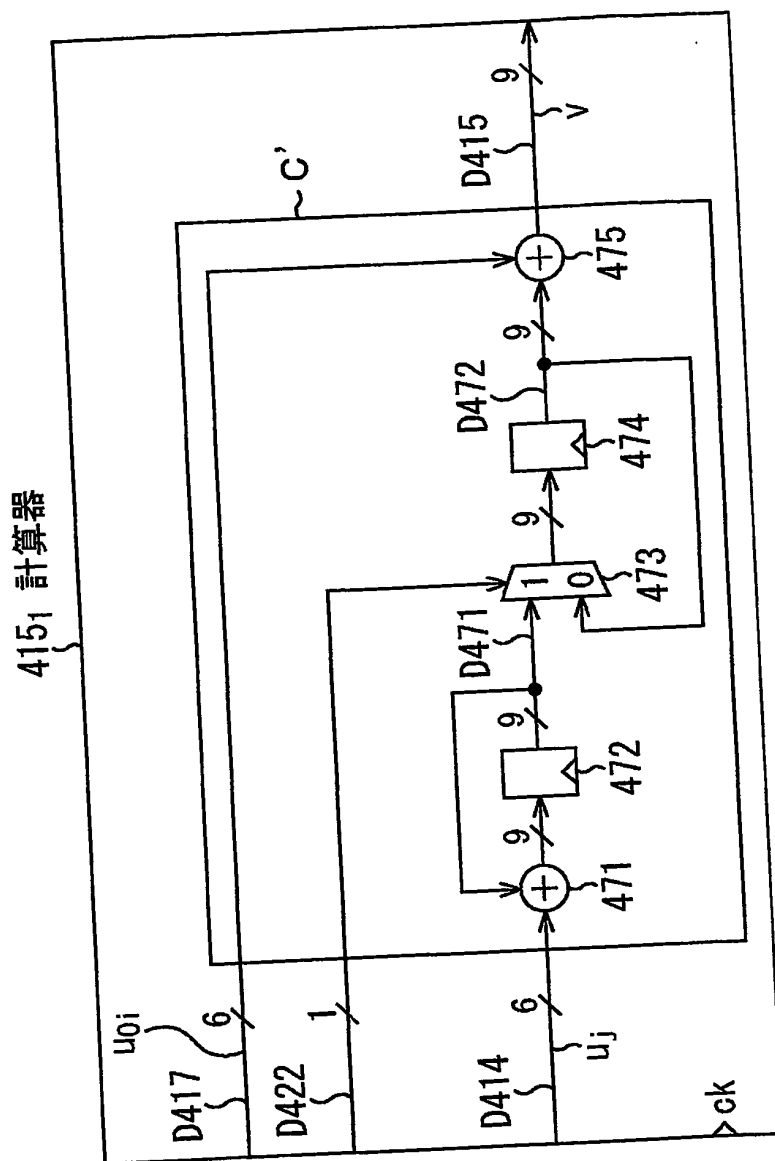


【図 21】  
図 21

412<sub>1</sub> 計算器



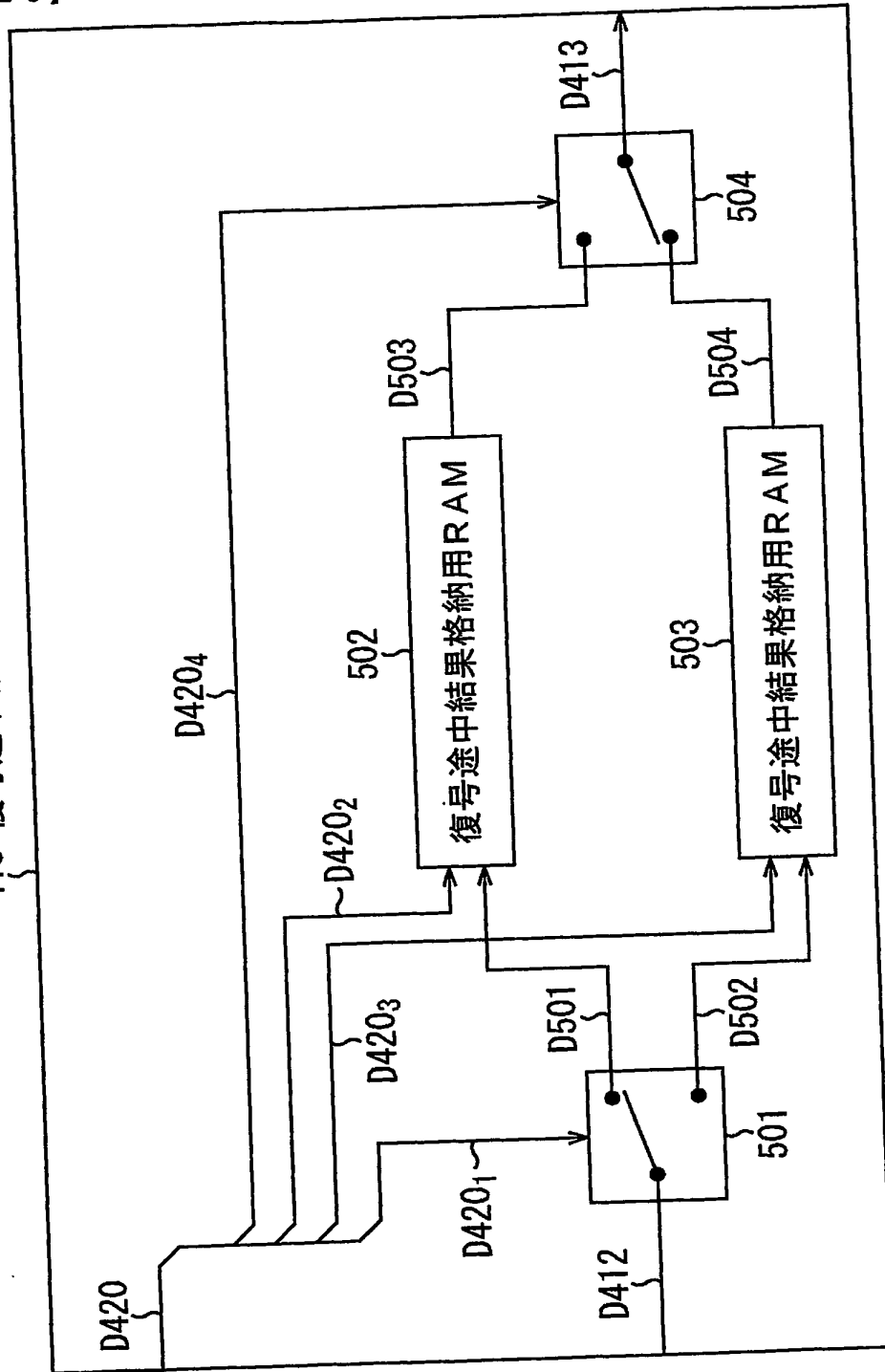
【図 22】  
図22



【図 23】

図23

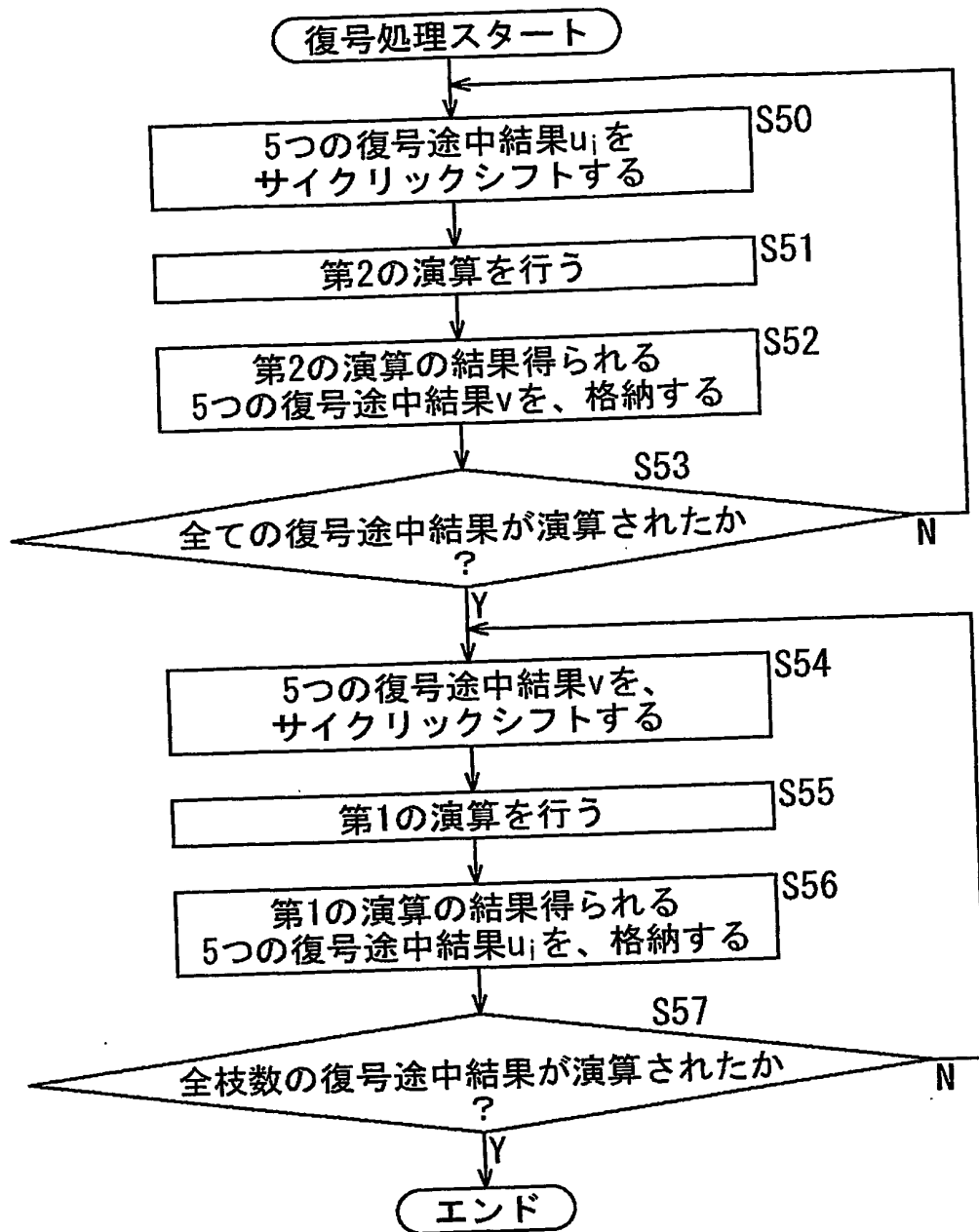
413 復号途中結果格納メモリ



【図 24】  
図24

復号途中結果格納用RAM502	第1～5行目の読み出し	第1～5行目の書き込み	第11～15行目の読み出し
復号途中結果格納用RAM503		第6～10行目の読み出し	第6～10行目の書き込み

↑  
t

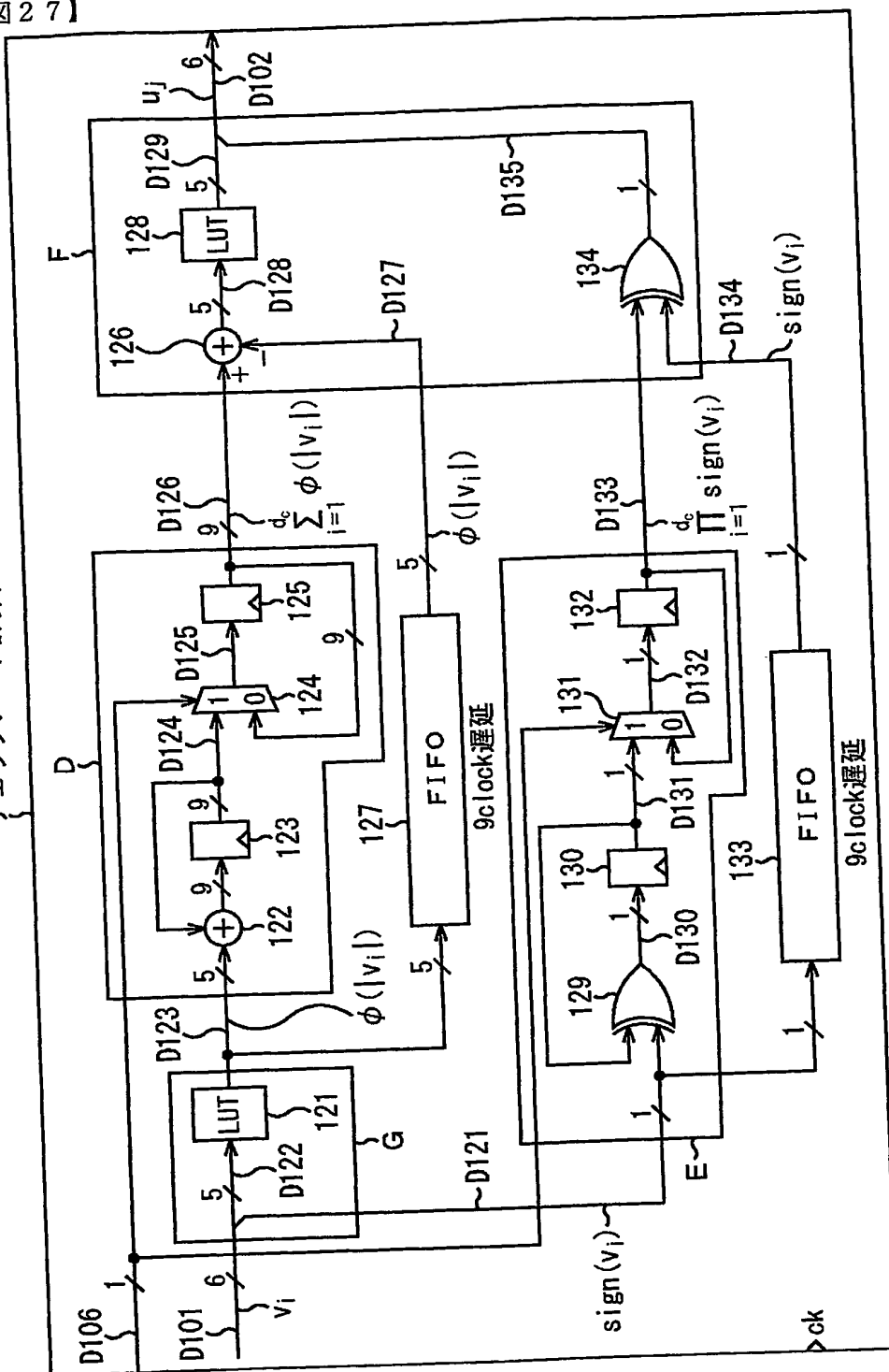
【図25】  
図25





【図 27】  
図 27

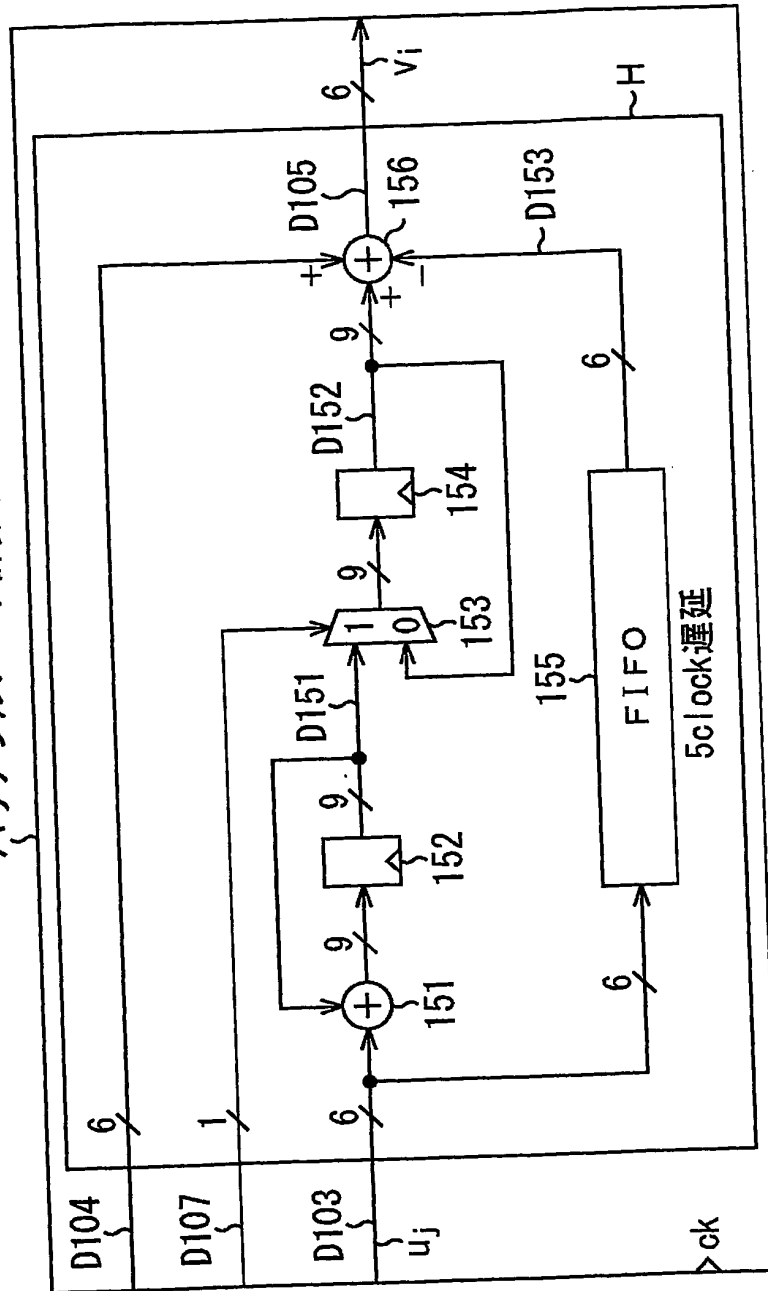
チェックノード計算器 101



【図 28】

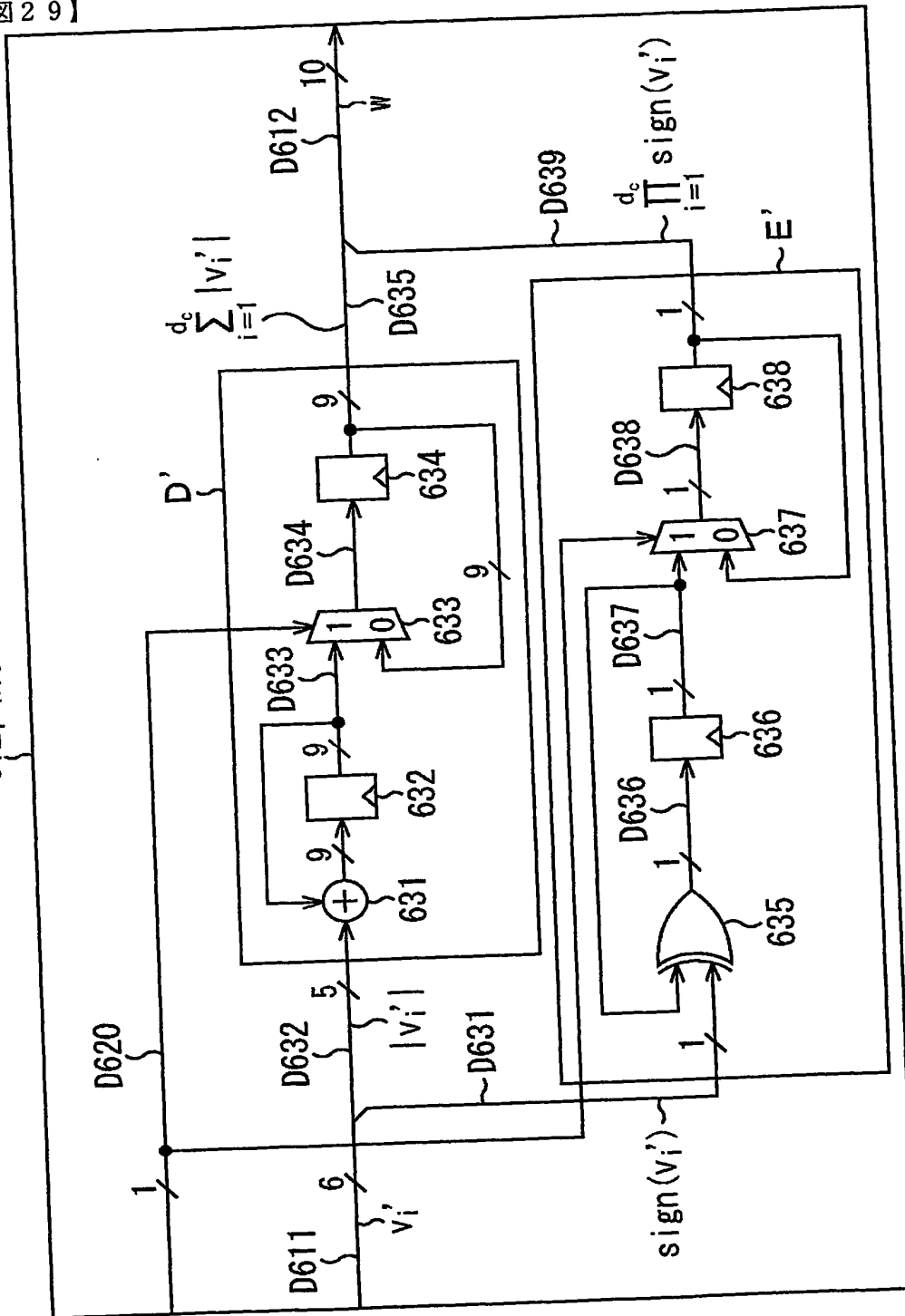
図28

バリアブルノード計算器 103



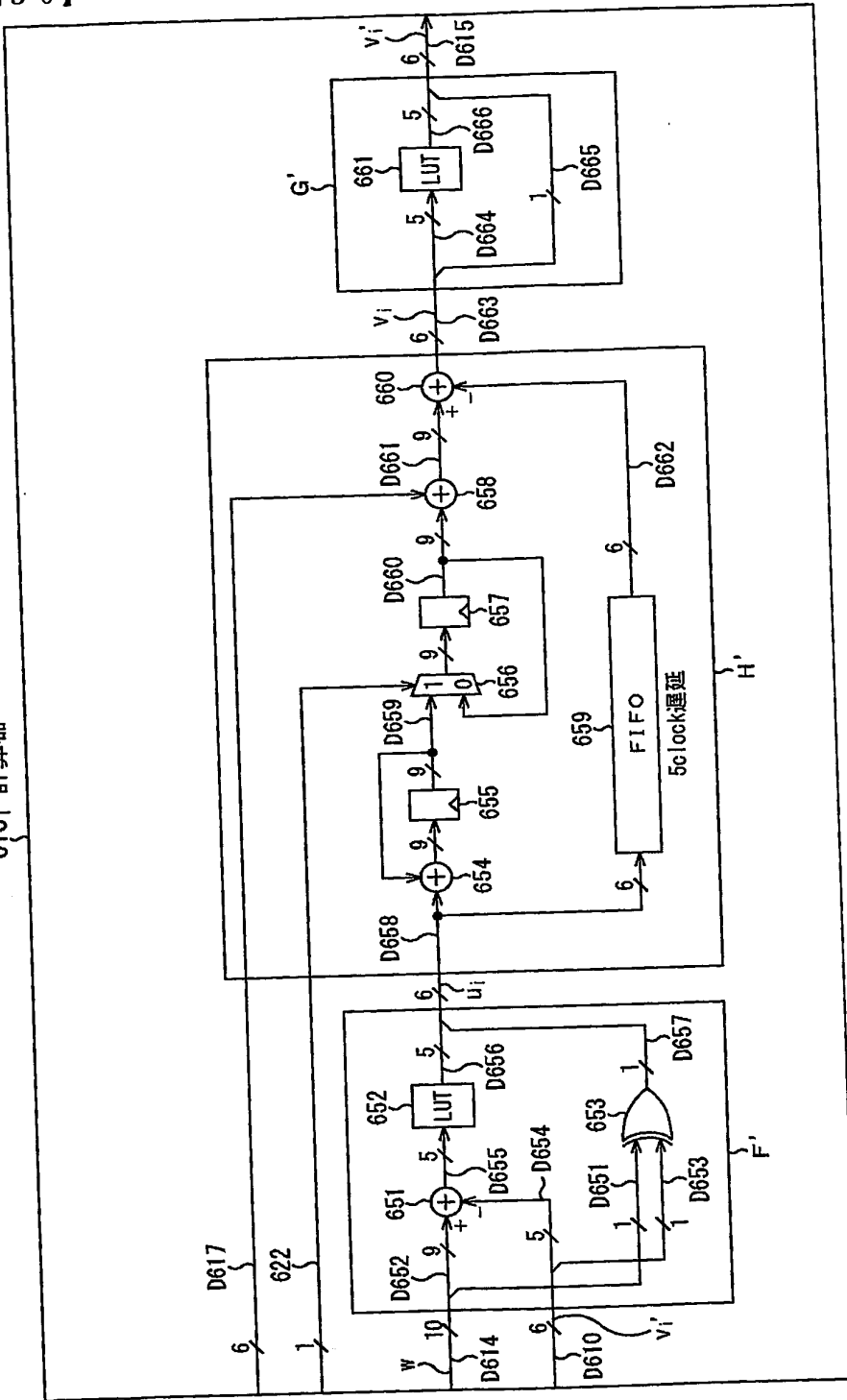
【図 29】  
図 29

6121 計算器



【図 30】  
図 30

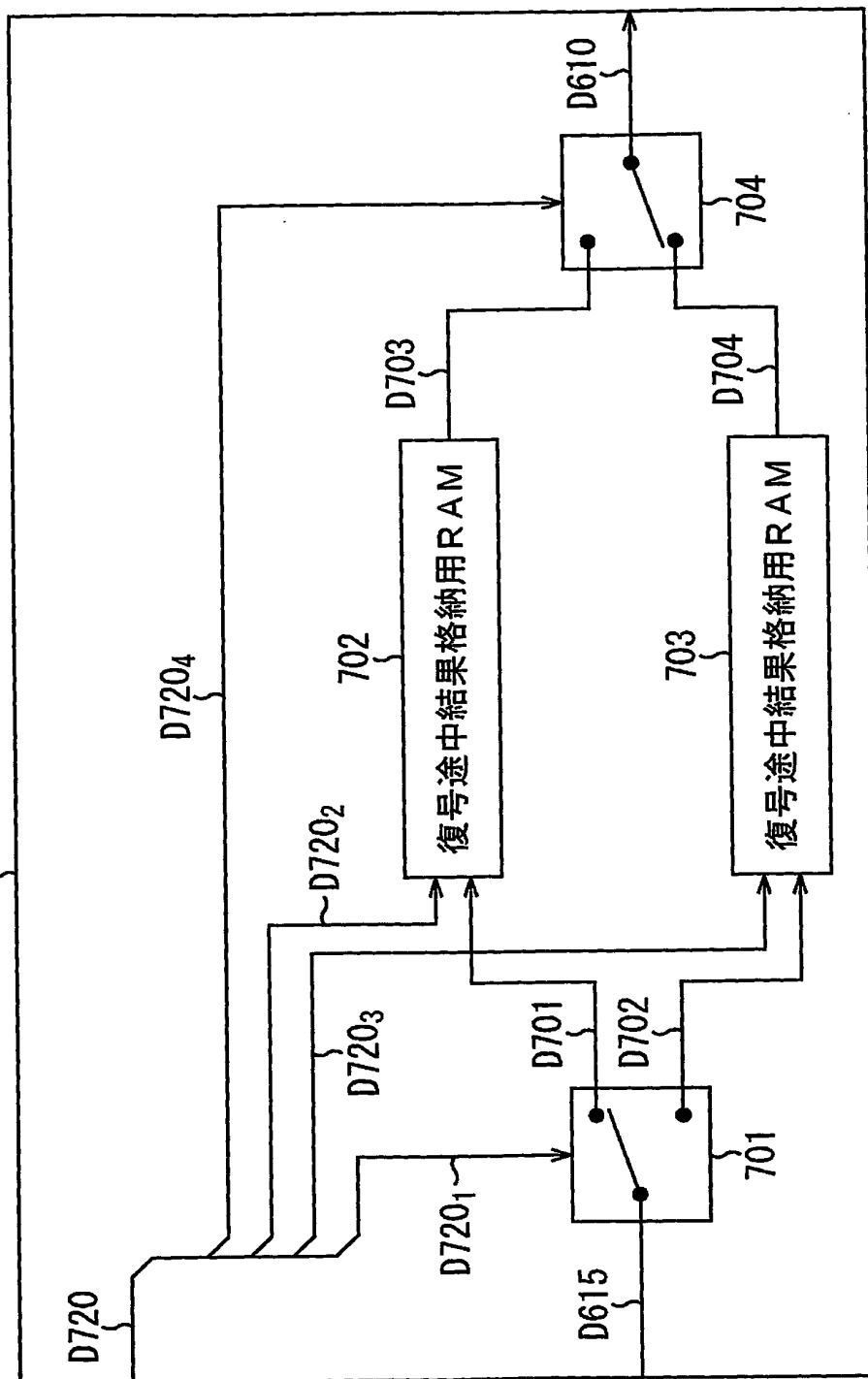
615<sub>1</sub> 計算器



【図 31】

図31

610 復号途中結果格納メモリ



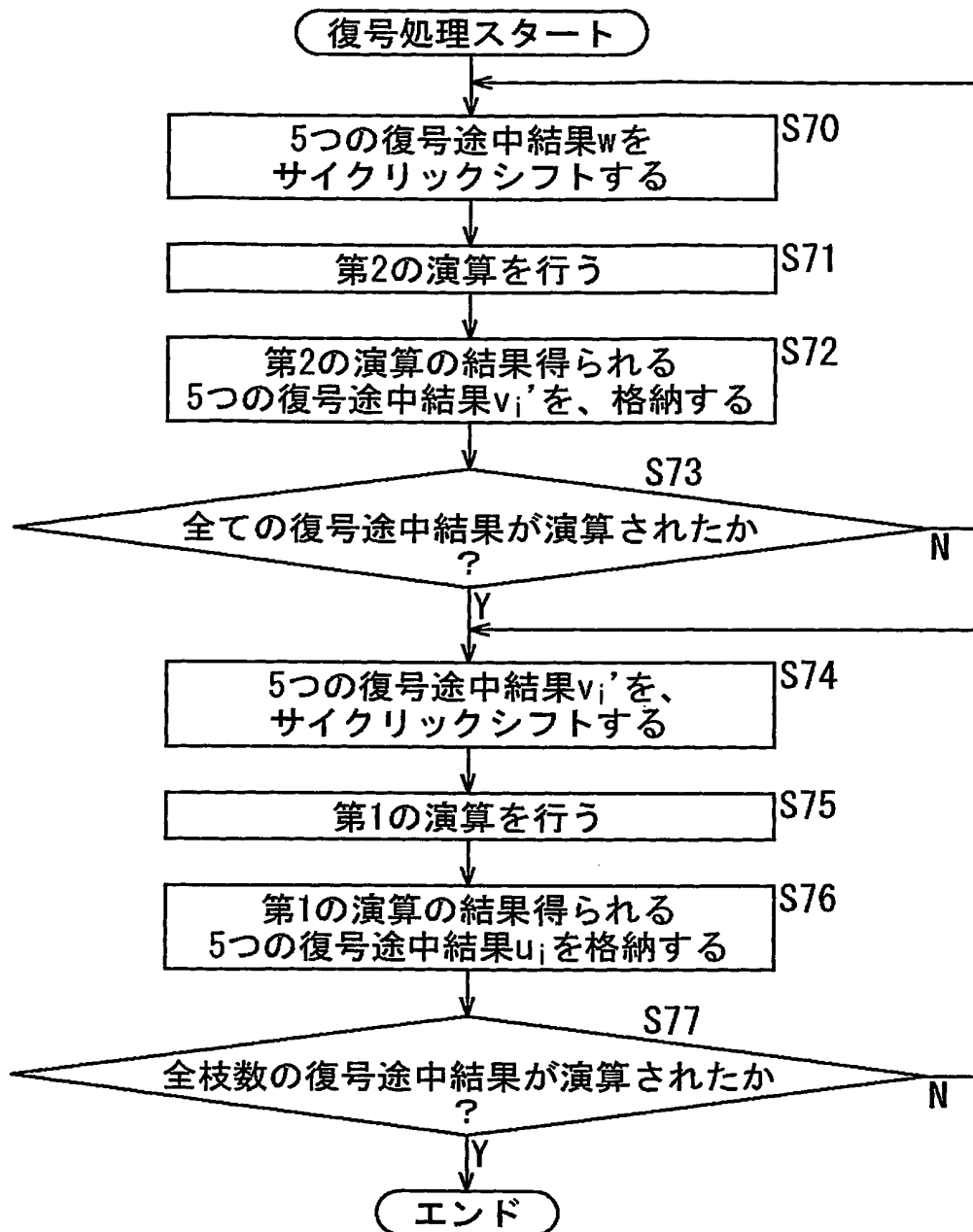
【図 32】  
図32

復号途中結果格納用RAM702	第1～5列目の読み出し	第1～5列目の書き込み	第11～15列目の読み出し
復号途中結果格納用RAM703		第6～10列目の読み出し	第6～10列目の書き込み

↑  
t

【図 33】

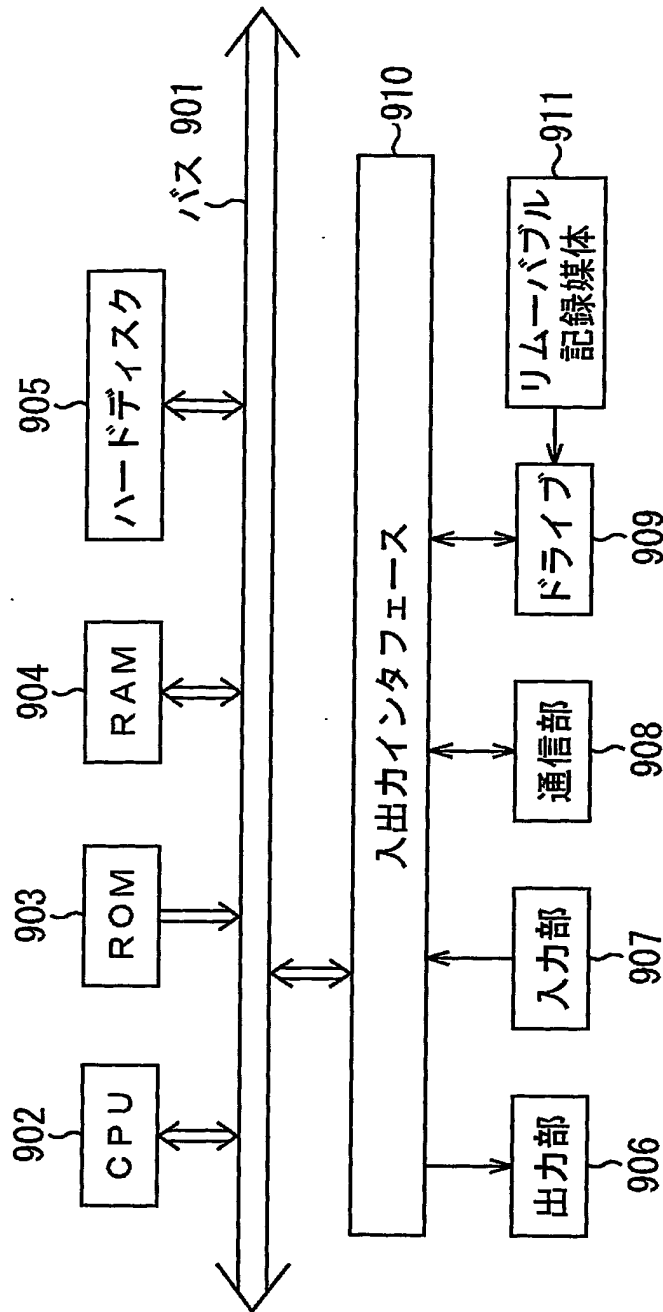
図33





【図 34】

図34



## 【書類名】 要約書

## 【要約】

## 【課題】

回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行うことができるLDPC符号の復号を実現する。

## 【解決手段】

LDPC符号の検査行列は、 $P \times P$ の単位行列、その単位行列の1のうちの1個から数個が0になった行列、それらのサイクリックシフト、それらの複数の和、 $P \times P$ の0行列の組合せで構成される。チェックノード計算部313は、チェックノードの演算を、 $P$ 個同時に行い、バリアブルノード計算部319は、バリアブルノードの演算を、 $P$ 個同時に行う。

## 【選択図】 図16

## 認定・付加情報

特許出願の番号	特願 2003-294383
受付番号	50301355781
書類名	特許願
担当官	第八担当上席 0097
作成日	平成15年 8月21日

## &lt;認定情報・付加情報&gt;

## 【特許出願人】

## 【識別番号】

000002185

## 【住所又は居所】

東京都品川区北品川6丁目7番35号

## 【氏名又は名称】

ソニー株式会社

## 【代理人】

申請人

## 【識別番号】

100082131

## 【住所又は居所】

東京都新宿区西新宿7丁目11番18号 711

ビルディング4階 稲本国際特許事務所

## 【氏名又は名称】

稲本 義雄

特願 2 0 0 3 - 2 9 4 3 8 3

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 2 1 8 5 ]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都品川区北品川 6 丁目 7 番 3 5 号

氏 名

ソニー株式会社

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**